(NASA-CR-167944) ENGINE DYNAMIC ANALYSIS WITH GENERAL NONLINEAR FINITE ELEMENT CODES. PART 2: BEARING ELEMENT IMPLEMENTATION OVERALL NUMERICAL CHARACTERISTICS AND BENCHMAKING (Akron Univ., Ohio.) 229 p

N82-33390

HC A11

Unclas

G3/07 35173

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR 167944		

4. Title and Subtitle	5. Report Date
Engine Dynamic Analysis with General Nonlinear Finite Element Codes Part II: Bearing Element Implementation, Overall Numerical Characteristics, and Benchmarking	October 1984
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
J. Padovan, M. Adams, J. Fertis, I. Zeid, and P. Lam	
	10. Work Unit No.
9. Performing Organization Name and Address	11. Contract or Grant No.
The University of Akron Department of Mechanical Engineering Akron, OH 44325	NSG 3283
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Topical
National Aeronautics and Space Administration Washington, DC 20546	14. Sponsoring Agency Code

15. Supplementary Notes Technical Monitor: C. C. Chamis,
NASA Lewis Research Center
Mail Stop 49-6
21000 Brookpark Road
Cleveland, OH 44135

16. Abstract
An interactive finite element, developed for turbine engine structures rotor/stator interaction in the first phase of this program, was implanted in a general purpose finite element computer code. Three different numerical integration methods were incorporated as part of the implantation in order to solve the governing structural dynamics equations of the interactive finite element. The general purpose code is used subsequently to predict the structural dynamic response of the rotor/stator coupled structure subjected to unbalance and impulse-type excitations. The structural-dynamic response predicted includes: (1) Bearing/rotor trajectories, (2) stator trajectory, (3) rotor orbit, and (4) force, velocity and acceleration histories at a given location in the coupled structure. Three-dimensional post-processors have been developed to graphically display the voluminous predicted results in isometric views.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Interactive finite element, bearing/rotor/stator, structural dynamics, numerical integration, graphical displays, computer programming, engine structures.	Unclassified, Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified		

* For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA CR-167944

ENGINE DYNAMIC ANALYSIS WITH GENERAL NONLINEAR FINITE ELEMENT CODES

PART II
BEARING ELEMENT IMPLEMENTATION, OVERALL NUMERICAL CHARACTERISTICS AND BENCHMARKING

By

J. Padovan

M. Adams

J. Fertis

I. Zeid

P. Lam

October 1981

Departments of Mechanical and Civil Engineering
The University of Akron
Akron, Ohio 44325

Prepared by
Lewis Research Center
National Aeronautics and Space Administration
Cleveland, Ohio 44315

NASA Grant NSG 3283

ABSTRACT

In an attempt to increase jet engine efficiencies, typically smaller rotor-stator-running clearances are being employed in current and new designs. Because of this and the ever present need to improve maintenance, reliability and structural integrity under various modes of operation, more sophisticated analyses tools are required to realistically model the engine. Due to the wide spread usage of general purpose finite element codes in the open market, this report seeks to adapt the procedure to use in modelling rotor-bearing-stator structure common to the turbine industry. The work outlined in this report covers the second phase of work on a three-phase NASA Lewis sponsored research grant on engine dynamic simulation by developing strategies which enable the use of available finite element codes. The second phase has concentrated on four major objectives, namely: i) Benchmarking the elements developed in Phase 1 by incorporation into a general purpose code (ADINA); ii) Evaluation of the numerical characteristics of finite element type rotor-bearing-stator simulations through the use of various types of currently available as well as specially developed explicit/implicit numerical integration operators; iii) Improving the overall numerical efficiency of the procedure; and, iv) Benchmarking the overall approach in several case studies.

INDEX

TABLE OF FIGURE CAPTIONS

1. INTRODUCTION

Due to the need to develop increased jet engine efficiencies, typically smaller rotor-stator running clearances are being employed in current and new designs. This together with the ever present thrust to improve reliability and reduce maintenance costs has generated the need to develop a better understanding of the basic dynamic characteristics of existing new engine configurations.

With the emergence of general purpose (GP) finite element (FE) codes such as ADINA, ANSYS, MARC, NASTRAN, [1,2] the analysis of the dynamics of general structures has been made readily available to structural designers and analysts. This follows from the fact that the overall FE methodology includes numerous important features namely [3,4]:

1) Handles hierarchal structural systems with numerous levels and substructure;
2) Handles large deformation kinematics-kinetics;
3) Accommodates general material properties including anisotropy, nonlinear elasticity, plasticity, viscoplasticity, etc.
4) Well suited to handle nonlinear boundary conditions as defined by impact, contact, rubbing, and;
5) There exist numerous commercially available GP codes with extensive element libraries [1,2].

Because of such capacities, GP FE codes are experiencing a wide spread usage throughout the various aerospace and commercial industries [1,2].

In spite of its far reaching capabilities, the FE procedure has not been extensively employed to analyze rotor-bearing-stator structure. This follows from the lack of elements which can model the interfacial fields generic to gas turbines and in general rotating equipment. Specifically, there is a lack of elements which can properly and efficiently model the interactive force fields originating in roller bearings, squeeze-film dampers, seals and rub/impact events, etc. Together with these shortcomings arises the difficulties associated with the integration of rotor-bearing-stator simulations which, unlike the modal approach say of Adams [5], requires the entire set of simulating equations to be handled directly. While Guyan type reduction schemes [6] can be employed to reduce the number of degrees of freedom of the various linear substructures, the direct integration of the coupled set of simulating equations still requires further work to improve numerical stability as well as computational efficiency. This is a direct outgrowth of the highly nonlinear manner in which various interfacial fields, such as those generated in squeeze-films and rubs, behave.

In the context of the foregoing, the basic emphasis of this three-year study is to develop as well as extend the FE methodology to efficiently handle the transient/steady state response of rotor-bearing-stator structure associated with gas turbine engines. Specially, this includes: (1) the development of specialized elements which enable the characterization of squeeze-film dampers, roller bearings, rubs, impacts, and seals [7]; (2) the benchmarking of the elements through incorporation into a GP code wherein direct integration schemes are used to generate the overall solution; and (3) the evaluation/improvement of the numerical operating characteristics of such simulations.

The thrust of this report is to outline the second year efforts aimed primarily at:

i) Implanting the squeeze-film damper element into a general purpose FE code (ADINA [8,9]) for testing and evaluation, including such items as:

   a). Element computational efficiency;

   b) Adequacy of the perturbational scheme in defining instantaneous stiffners and damping coefficients of fluid film; and,

   c) Adequacy of difference mesh [7] used in squeeze-film element to model interactive forces.

ii) Determine running characteristics of direct integration approach of FE generated rotor-bearing-stator simulation, including:

   a) Benchmarking scheme;

   b) Establish/improving numerical efficiency [10,11];

   c) Establish numerical stability threshold;

   d) Comparison of explicit [12] vs implicit [13] methodologies of direct integration; and

   e) Determine extent and source of nonlinearity arising in model.

In the section which follows, detailed discussions are given on the overall implant strategy, the governing field equations, the explicit and implicit integration schemes as well as the results of numerical experiments used to benchmark the operating characteristics of the approach. The appendicies include listings of the appropriate coding required to implant the bearing element into the ADINA architecture as well as its associated user instructions. Also included is a listing of complementing graphics post processors enabling various views of rotor trajectories.

2. OVERALL APPROACH

Rotor-squeeze film/bearing-stator simulations are inherently nonlinear. This is an outgrowth of two sources namely:

i) The inherent behavior characteristic to squeeze films particularly during transient situations, and

ii) Potential structural interactions wherein either large deformation kinematics or material nonlinearity (plasticity) are excited.

Because of such features, the overall rotor-bearing-stator simulation must be able to incorporate the various sources of nonlinearity. Additionally, to enable efficient solutions in situations wherein various of the model components are linear, the overall simulation scheme must incorporate substructuring capabilities. Furthermore, since transient problems will be considered, such features must be accommodated by the various integration algorithms used to solve the governing model equations.

The central issue of the modelling requirements is the need to identify the separate sources of nonlinearity which may occur. Having done so, such sources can be handled via substructural procedures. Since numerical integration is used to solve the resulting set, the various linear components of the model can all be generated and stored before time stepping occurs. In this context, only the nonlinear substructural components need be updated during integration. After this, the various linear substructural contributions can be assembled into the global scheme. This is the approach taken for this study.

Figures 2-1 and 2-2 illustrate the various aspects associated with the overall flow of control for the foregoing substructural approach. Specifically, Figure 1-1 illustrates the basic flow of calculations for

explicit type schemes. As can be seen three basic loops are preset in the logic flow. These are associated with the linear and nonlinear substructural components as well as the explicit integration process itself. For linear structural components, the substructural stiffness is calculated and globally assembled outside the integration loop. During integration, the reaction forces of such components are obtained simply by post-multiplying the linear stiffness by the appropriate current deflection. This will be discussed in detail in Chapter 4. Such an approach significantly streamlines the calculation flow. For the nonlinear loop, the nodal load due to component reactions must be updated and reassembled during each time step of the explicit integration scheme. This also applies to the bearing group.

For the implicit integration scheme, as seen from Figure 2-2, the overall flow is similar to that of the explicit case but requires the additional calculation of stiffness and damping matrices associated with the linear, nonlinear and bearing components. Again, the linear component stiffness and damping matrices are generated and globally assembled outside the integration loop while the nonlinear and bearing elements are constantly updated within the loop.

3. GOVERNING FINITE ELEMENT FIELD EQUATIONS

Assuming the possibility of large deformation kinematics and kinetics, the Lagrangian type equations of motion take the form [14].

$$\frac{\partial}{\partial a_j}\left(S_{jk}\left(\delta_{ik}+\frac{\partial u_i}{\partial a_k}\right)\right)+\rho_o F_{oi}=\rho_o u_{i,tt} \tag{3.1}$$

where a_i; $i = 1,2,3$ are the cartesian coordinates defining the initial

configuration, S_{jk} the 2nd Piola-Kirchhoff stress tensor, δ_{ik} the Kronecker delta, ρ_0 the initial density, u_i the Lagrangian displacement components, F_{oi} the body force and $(\ )_{,tt}$ denotes time differentiation. Based on (3.1), the virtual work principle used to establish the requisite FE formulation is given by [14]

$$\int_R (\delta L_{ij} S_{ij} + \delta u_i \rho_0 (u_{i,tt} + F_{oi}))dv = \int_{\partial R} \delta u_i n_j S_{jk} (\delta_{ik} + \frac{\partial u_i}{\partial a_k})\, ds \quad (3.2)$$

where $\delta(\ )$ is the virtual operator, ∂R the boundary of R, n_i the normal to ∂R and L_{ij} the Lagrangian strain tensor is defined by [14].

$$L_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i} + u_{\ell,i} u_{\ell,j}) \quad (3.3)$$

Employing the usual shape function approximation we have that [3]

$$\underset{\sim}{u}(\underset{\sim}{a},t) = [N(\underset{\sim}{a})]\underset{\sim}{Y}(t) \quad (3.4)$$

where $[N(\underset{\sim}{a})]$ is the shape function, $\underset{\sim}{Y}$ the nodal displacement and

$$\underset{\sim}{u}^T = (u_1, u_2, u_3) \quad (3.5)$$

In terms of (3.4), L_{ij} can be recast as [3]

$$\underset{\sim}{L} = [[B] + \frac{1}{2}[B_n(\underset{\sim}{Y})][G]]\underset{\sim}{Y} \quad (3.6)$$

where $[B]$, $[B_n(\underset{\sim}{Y})]$ and $[G]$ are defined in Appendix A7 and

$$\underset{\sim}{L}^T = (L_{11}, L_{22}, L_{33}, L_{12}, L_{23}, L_{31}) \quad (3.7)$$

Substituting (3.4) and (3.6) into (3.2) yields the following FE formulation of the virtual work principle namely

$$\int_R (\delta \underset{\sim}{L}^T \underset{\sim}{S} + \delta \underset{\sim}{U}^T \rho_0 \underset{\sim}{u}_{,tt} + \rho_0 \underset{\sim}{F}_0)dv = \delta \underset{\sim}{Y}^T \underset{\sim}{F} \quad (3.8)$$

where

$$S^T = (S_{11}, S_{22}, S_{33}, S_{12}, S_{23}, S_{13}) \tag{3.9}$$

$$\delta YF = \int_{\partial R_e} \delta u_i n_j S_{jk} \left(\delta_{ik} + \frac{\partial u_i}{\partial a_k}\right) ds \tag{3.10}$$

$$\delta L = [B^*]\delta Y$$

$$= [[B] + [B_N][G]]\delta Y \tag{3.11}$$

Now, since δY is arbitrary, (3.8) reduces to the following expression for a given e^{th} element namely

$$\int_{R_e} (\rho_0[N]^T[N]\ddot{Y} + [B^*]^T S + \rho_0[N]\Gamma_0)dv = F_e \tag{3.12}$$

Note here F_e strictly defines the elements contribution to its attachment nodes. Recasting (3.12) in more convenient matrix form yields the following expression namely

$$[M_e]\ddot{Y} + \int_{R_e} [B^*]^T S dv + f_e = F_e \tag{3.13}$$

where the consistent mass matrix [11] $[M_e]$ and f_e the body force matrix are given by

$$[M_e] = \int_{R_e} \rho_0[N]^T[N]dv \tag{3.14}$$

$$f_e = \int_{R_e} \rho_0[N]^T \Gamma_0 dv \tag{3.15}$$

To develop an overall simulation of the rotor-stator support structure, (3.13) must be assembled together with the interactive force representation [7] of the squeeze film damper and roller bearing combinations. This can be achieved in two different ways evolving out of the requirements of explicit or implicit integration.

ORIGINAL PAGE IS
OF POOR QUALITY

For the explicit formulation [12], the interacting fluid forces of the squeeze film can be modeled as nodal loads acting on the various appropriate rotor-stator nodes. This type of formulation takes the following assembled form namely

$$[M]\ddot{\tilde{Y}} + \int_R [B^*]^T \tilde{S} dv = - \tilde{F}_b + \tilde{f} + \tilde{F}_{ext} \tag{3.16}$$

where $\tilde{F}_{ext}$ denotes potential externally applied nodal loads, $\tilde{F}_b$ is the interacting squeeze film load, and f the body force is given by

$$\tilde{f} = \int_R \rho_o [N] \tilde{f}_o dv \tag{3.17}$$

For this study the loads developed in the squeeze film are simulated in the manner described in Report 1 [7]. Specifically, two alternatives can be employed:

i) The pressure profiles modelled by the Reynolds simulation are integrated over the entire squeeze film bearing surface to yield a single force vector defining the overall force occurring in the fluid film; or,

ii) The profiles can be integrated over segments of bearing arc to yield a detailed set of loads defining squeeze film forces, wherein a more detailed analysis admitting deformations of the bearing structure can be accommodated.

For implicit type formulations [13] (3.16) must be expanded in Taylor series retaining solely up to the tangent stiffness and damping expressions. Specifically, if we let

$$\tilde{Y}(t+\Delta t) \sim \tilde{Y}(t) + \Delta\tilde{Y}(t+\Delta t) \tag{3.18}$$

$$\dot{\tilde{Y}}(t+\Delta t) \sim \dot{\tilde{Y}}(t) + \Delta\dot{\tilde{Y}}(t+\Delta t) \tag{3.19}$$

such that

$$||\underset{\sim}{Y}(t)|| >> ||\Delta\underset{\sim}{Y}(t+\Delta t)|| \tag{3.20}$$

$$||\dot{\underset{\sim}{Y}}(t)|| >> ||\Delta\dot{\underset{\sim}{Y}}(t+\Delta t)|| \;, \tag{3.21}$$

then the proper expansions can be taken namely

$$\underset{\sim}{L}(t+\Delta t) \sim \underset{\sim}{L}(t) + \Delta\underset{\sim}{L}(t+\Delta t) \tag{3.22}$$

$$\underset{\sim}{S}(t+\Delta t) \sim \underset{\sim}{S}(t) + \Delta\underset{\sim}{S}(t+\Delta t) \tag{3.23}$$

$$\underset{\sim}{F}_b(t+\Delta t) \sim \underset{\sim}{F}_b(t) + \Delta\underset{\sim}{F}_b(t+\Delta t) \tag{3.24}$$

where

$$\Delta\underset{\sim}{L}(t+\Delta t) = [B^*(\underset{\sim}{Y}(t))]\Delta\underset{\sim}{Y}(t+\Delta t) \tag{3.25}$$

$$\Delta\underset{\sim}{S}(t+\Delta t) = [D(\underset{\sim}{Y}(t))]\Delta L(t+\Delta t) \tag{3.26}$$

$$\Delta\underset{\sim}{F}_b(t+\Delta t) = [K_b(\underset{\sim}{Y}(t))]\Delta\underset{\sim}{Y}(t+\Delta t) + [C_b(\underset{\sim}{Y}(t), \dot{\underset{\sim}{Y}}(t))]\Delta\dot{\underset{\sim}{Y}}(t+\Delta t) \tag{3.27}$$

such that [D] is the tangent stiffness of the structural partitions of the rotor-stator assembly while $[K_b]$ and $[C_b]$ are the pseudo stiffness and damping matrices developed across the squeeze film [7]. In terms of (3.18 – 3.27), (3.16) can be reduced to the following tangent formulation that is

$$[M]\ddot{\underset{\sim}{Y}}(t+\Delta t) + [C_b]\Delta\dot{\underset{\sim}{Y}}(t+\Delta t) + [[K] + [K_b]]\Delta\underset{\sim}{Y}(t+\Delta t) = - \underset{\sim}{F}_b(t) + \underset{\sim}{f}(t+\Delta t) - \int_R [B^*]^T \underset{\sim}{S}\Big|_t dv \tag{3.28}$$

where

$$[K] = \int_R ([G]^T[S(t)][G] + [B^*]^T[D(t)][B^*])dv \tag{3.29}$$

such that [S(t)] is a matrix of stress defined at time t.

Note for purely elastic structural components, [D] defines the so called tangent stiffness which is the slope of the stress-strain space. In the case where some structural components may have undergone some plastic deformation, [D] is replaced by $[D_{ep}]$ which is the elastic-plastic counterpart. A good description of the derivation of $[D_{ep}]$ is given by Zienkiewicz [3]. In such situations, (3.28) is replaced by

$$[M]\ddot{\underset{\sim}{Y}} + [C_b]\Delta\dot{\underset{\sim}{Y}} + [[K_{ep}] + [K_b]]\Delta Y =$$

$$- \underset{\sim}{F}_b(t) + \underset{\sim}{f}(t + \Delta t) - \int_R [B^*]^T \underset{\sim}{S}\big|_t dv \tag{3.30}$$

such that

$$[K_{ep}] = \int_R ([G]^T[S(t)][G] + [B^*]^T[D_{ep}(t)][B^*])dv \tag{3.31}$$

Note for situations wherein purely linear structural components are involved, (3.28) reduces to the form

$$[M]\ddot{\underset{\sim}{Y}}(t + \Delta t) + [C_b]\Delta\dot{\underset{\sim}{Y}}(t + \Delta t) + [[K_\ell] + [K_b]]\Delta\underset{\sim}{Y}(t + \Delta t) =$$

$$- \underset{\sim}{F}_b(t) + \underset{\sim}{f}(t + \Delta t) - \int_R [B]^T \underset{\sim}{S}\big|_t dv \tag{3.32}$$

where here

$$[K_\ell] = \int_R [B]^T[D][B]dv \tag{3.33}$$

For such situations, $[K_\ell]$ needs to be updated solely at the start of the problem.

4. SOLUTION METHODOLOGY

The main difficulties in the rotor-stator simulations evolve out of the fact that during transient occurrences, the interactive squeeze film forces undergo significant changes in direction and magnitude. This trend

tends to introduce rather strong changes in $[K_b]$ and $[C_b]$ as the calculations proceed from time step to time step. The major result of this is that a perfusion of system harmonics are then introduced in the problem hence reducing the stability threshold of implicit integration schemes. Since explicit schemes tend to be more stable in problems involving a perfusion of system harmonics, as noted earlier, we have adopted the dual approach of employing both explicit and implicit schemes of integration. This enables us to strike a balance between solution efficiency versus numerical stability.

For the current purposes, we have adopted the central difference form of explicit operator [15]. In this context, since [16]

$$\ddot{\underset{\sim}{Y}} = \frac{1}{(\Delta t)^2}(\underset{\sim}{Y}(t + \Delta t) - 2\underset{\sim}{Y}(t) + \underset{\sim}{Y}(t - \Delta t)) \tag{4.1}$$

the following explicit time stepping scheme can be developed to solve (3.16) namely

$$\underset{\sim}{Y}(t + \Delta t) = 2\underset{\sim}{Y}(t) - \underset{\sim}{Y}(t - \Delta t) + (\Delta t)^2[M]^{-1}(\underset{\sim}{f}\Big|_t + \underset{\sim}{F}_{ext}\Big|_t - \underset{\sim}{F}_b\Big|_t - \int_R [B^*]^T \underset{\sim}{S}\Big|_t dv) \tag{4.2}$$

Note in (4.2), $\underset{\sim}{F}_b\Big|_t$ and $\int_R [B^*]^T \underset{\sim}{S}\Big|_t dv$ are updated each time step.

For rotor-stator structure involving purely linear media and small deformation kinematics, the following simplification can be employed that is

$$\int_R [B^*]^T \underset{\sim}{S}\Big|_t dv = \int_R [B]^T[D][B]dv\underset{\sim}{Y}(t) \tag{4.3}$$

where [D] is the material stiffness associated with the typical Hookean constitutive law. In terms of (4.3), (4.2) reduces to the following form

$$\underset{\sim}{Y}(t + \Delta t) = 2\underset{\sim}{Y}(t) - \underset{\sim}{Y}(t - \Delta t) + (\Delta t)^2[M]^{-1}(\underset{\sim}{f}\Big|_t + \underset{\sim}{F}_{ext}\Big|_t - \underset{\sim}{F}_b\Big|_t - [K_\ell]\underset{\sim}{Y}(t)) \tag{4.4}$$

where $[K_\ell]$ is the constant stiffness matrix defined by

$$[K_\ell] = \int_R [B]^T[D][B]dv \tag{4.5}$$

As can be seen from (4.4), the only term requiring updating on the right-hand side is $\underset{\sim}{F}_b$ the interactive squeeze-film force. Hence, apart from requiring small Δt, (4.4) represents an architecturally very streamlined algorithm to employ in rotor-stator simulations.

For situations wherein the rotor-stator structure involves both linear and nonlinear components, the overall calculations can be substructured In this context

$$\int_R [B^*]^T\underset{\sim}{S}\Big|_t dv = \int_{R_N} [B^*]^T\underset{\sim}{S}dv + \int_{R_\ell} [B]^T[D][B]dv\ \underset{\sim}{Y}(t) \tag{4.6}$$

where

$$R = R_\ell + R_N \tag{4.7}$$

such that R_N and R_ℓ are the regions defining the linear and nonlinear component zones respectively. In the context of (4.6), (4.2) can be recast as follows namely

$$\underset{\sim}{Y}(t + \Delta t) = 2\underset{\sim}{Y}(t) - \underset{\sim}{Y}(t - \Delta t) + (\Delta t)^2[M]^{-1}(\underset{\sim}{f}\Big|_t + \underset{\sim}{F}_{ext}\Big|_t - \underset{\sim}{F}_b\Big|_t - \int_{R_N} [B^*]^T\underset{\sim}{S}\Big|_t dv - [\bar{K}_\ell]\underset{\sim}{Y}(t)) \tag{4.8}$$

where here

$$[\bar{K}_\ell] = \int_{R_\ell} [B]^T[D][B]dv \tag{4.9}$$

The overall flow of calculations associated with the foregoing explicit scheme is given in Figure 4.1.

For the implicit scheme, the Wilson θ [16] and Newmark β [17] methods will be employed to integrate the tangent formulation. To start the development of the requisite algorithms, (3.28) is recast in a more appropriate form namely

$$[M]\underset{\sim}{\ddot{Y}}(t+\tau) + [C_b]\underset{\sim}{\dot{Y}}(t+\tau) + [[K] + [K_b]]\Delta\underset{\sim}{Y}(t+\tau) =$$

$$- \underset{\sim}{F}_b(t) + [C_b]\underset{\sim}{\dot{Y}}(t) + \underset{\sim}{F}(t+\tau) - \int_R [B^*]\underset{\sim}{S}\big|_t dv \tag{4.10}$$

or more simply

$$[M]\underset{\sim}{\ddot{Y}}(t+\tau) + [C_b]\underset{\sim}{\dot{Y}}(t+\tau) + [K^*]\Delta\underset{\sim}{Y}(t+\tau) = \underset{\sim}{F}^*(t+\tau) \tag{4.11}$$

where

$$[K^*] = [K] + [K_b] \tag{4.12}$$

$$\underset{\sim}{F}^*(t+\tau) = - \underset{\sim}{F}_b(t) + [C_b]\underset{\sim}{\dot{Y}}(t) + \underset{\sim}{F}(t+\tau) - \int_R [B^*]\underset{\sim}{S}\big|_t dv \tag{4.13}$$

Considering the Wilson θ method first, using the usual linear acceleration assumption, it follows that the velocity at $t + \Delta t$ takes the form

$$\underset{\sim}{\dot{Y}}(t+\tau) = \underset{\sim}{\dot{Y}}(t) + \frac{\tau}{2}(\underset{\sim}{\ddot{Y}}(t+\tau) + \underset{\sim}{\ddot{Y}}(t)) \tag{4.14}$$

similarly

$$\underset{\sim}{Y}(t+\tau) = \underset{\sim}{Y} + \tau\underset{\sim}{\dot{Y}}(t) + \frac{\tau^2}{6}(\underset{\sim}{\ddot{Y}}(t+\tau) + 2\underset{\sim}{\ddot{Y}}(t)) \tag{4.15}$$

Employing (4.14, 4.15) in conjunction with (4.10) yields the following expression for $Y(t+\tau)$ namely

$$\Delta\underset{\sim}{Y}(t+\tau) = [K_d]^{-1}\{\underset{\sim}{F}^*(t+\tau) + [M](\frac{6}{\tau}\underset{\sim}{\dot{Y}}(t) +$$

$$2\underset{\sim}{\ddot{Y}}(t) + [C_b](2\underset{\sim}{\dot{Y}}(t) + \frac{\tau}{2}\underset{\sim}{\ddot{Y}}(t)) - [K_d]\underset{\sim}{Y}(t)\} \tag{4.16}$$

where

$$[K_d] = [\frac{6}{\tau^2}[M] + \frac{3}{\tau}[C_b] + [K] + [K_b]] \tag{4.17}$$

Interpolating in the manner of Wilson [16], we let $\tau = \theta\Delta t (\theta > 1)$ so that

$$\ddot{Y}(t+\Delta t) = (1 - \frac{1}{\theta})\ddot{Y}(t) + \frac{1}{\theta}\ddot{Y}(t+\tau) \tag{4.18}$$

$$\dot{Y}(t+\Delta t) = \dot{Y}(t) + \frac{\Delta t}{2}(\ddot{Y}(t) + \ddot{Y}(t+\tau)) \tag{4.19}$$

$$Y(t+\Delta t) = Y(t) + \Delta t\dot{Y}(t) + \frac{(\Delta t)^2}{6}(\ddot{Y}(t+\Delta t) + 2\ddot{Y}(t)) \tag{4.20}$$

Note, as per Wilson [16], to yield a stable solution $\theta \geq 1.37$ must be employed. For our purposes, the value of $\theta = 1.4$ is used. In employing equations (4.16) and (4.18 - 4.20), it follows that the dynamic stiffness matrix $[K_d]$ must be updated and reinverted each time either $[K]$, $[K_b]$ or $[C_b]$ is reevaluated.

To establish the requisite algorithms for the Newmark method [17], the following approximations on velocity and displacement are employed namely

$$Y(t+\Delta t) = Y(t) + \Delta t\dot{Y}(t) + \frac{(\Delta t)^2}{2}\ddot{Y}(t) + \alpha(\Delta t)^2(\ddot{Y}(t+\Delta t) - \ddot{Y}(t)) \tag{4.21}$$

and

$$\dot{Y}(t+\Delta t) = \dot{Y}(t) + \Delta t\ddot{Y}(t) + \delta\Delta t(\ddot{Y}(t+\Delta t) - \ddot{Y}(t)) \tag{4.22}$$

such that for linear situations, it has been found that for unconditional stability, $\delta \geq \frac{1}{2}$ and $\alpha \geq \frac{1}{4}(\delta + \frac{1}{2})^2$. Solving (4.21) and (4.22) for $\ddot{Y}(t+\Delta t)$ and $\dot{Y}(t+\Delta t)$ in terms of $\Delta Y(t+\Delta t)$ yields

$$\ddot{Y}(t+\Delta t) = \ddot{Y}(t) + \frac{1}{\alpha(\Delta t)^2}(\Delta Y(t+\Delta t) - \Delta t\dot{Y}(t) - \frac{(\Delta t)^2}{2}\ddot{Y}(t)) \tag{4.23}$$

and

$$\Delta\underset{\sim}{\dot{Y}}(t+\Delta t) = \Delta t\underset{\sim}{\ddot{Y}}(t) + \frac{\delta}{\alpha\Delta t}\left(\Delta\underset{\sim}{Y}(t+\Delta t) - \Delta t\underset{\sim}{\dot{Y}}(t) - \frac{(\Delta t)^2}{2}\underset{\sim}{\ddot{Y}}(t)\right) \tag{4.24}$$

To obtain $\Delta\underset{\sim}{Y}(t+\Delta t)$, employing (4.23 and 4.24) in conjunction with (4.10) yields

$$\begin{aligned}
&[M]\left(\underset{\sim}{\ddot{Y}}(t) + \frac{1}{\alpha(\Delta t)^2}\left(\Delta\underset{\sim}{Y}(t+\Delta t) - \Delta t\underset{\sim}{\dot{Y}}(t) - \frac{(\Delta t)^2}{2}\underset{\sim}{\ddot{Y}}(t)\right)\right) + \\
&\quad [C_b]\left(\Delta t\underset{\sim}{\ddot{Y}}(t) + \frac{\delta}{\alpha\Delta t}\left(\Delta\underset{\sim}{Y}(t+\Delta t) - \Delta t\underset{\sim}{\dot{Y}}(t) - \frac{(\Delta t)^2}{2}\underset{\sim}{\ddot{Y}}(t)\right)\right) + \\
&\quad [K^*]\Delta\underset{\sim}{Y}(t+\Delta t) = \underset{\sim}{F}^*(t+\Delta t)
\end{aligned} \tag{4.25}$$

Solving for $\Delta\underset{\sim}{Y}(t+\Delta t)$ we have that

$$\begin{aligned}
&\Delta\underset{\sim}{Y}(t+\Delta t) = \\
&\quad \left[\frac{1}{\alpha(\Delta t)^2}[M] + \frac{\delta}{\alpha\Delta t}[C_b] + [K^*]\right]^{-1}\Big\{\underset{\sim}{F}^*(t+\Delta t) - \\
&\quad [M]\left(\underset{\sim}{\ddot{Y}}(t) - \frac{1}{\alpha(\Delta t)^2}\left(\Delta t\dot{Y}(t) + \frac{(\Delta t)^2}{2}\ddot{Y}(t)\right)\right) - \\
&\quad [C_b]\left(\Delta t\underset{\sim}{\ddot{Y}}(t) - \frac{\delta}{\alpha\Delta t}\left(\Delta t\underset{\sim}{\dot{Y}}(t) + \frac{(\Delta t)^2}{2}\underset{\sim}{\ddot{Y}}(t)\right)\right)\Big\}
\end{aligned} \tag{4.26}$$

As with the Wilson scheme, it follows that the dynamic stiffness appearing in (4.26) must be updated and reinverted each time either [K], $[K_b]$ or $[C_b]$ is reevaluated. This is clearly seen from the overall flow of control depicted in Figure 4.2.

5. ADINA BASED BEARING IMPLANT AND ASSOCIATED GRAPHICS POST PROCESSORS

To benchmark the approach developed sections 2-4 the appropriate general purpose finite element code had to be selected. As noted in the first year report [7], the ADINA code was chosen for this purpose. This follows from the fact that it has the requisite features required for rotor-bearing-stator simulations. These include [7,9]:

i) Linear and nonlinear substructuring features;

ii) An extensive element library;

iii) Capability to handle kinetic, kinematic, and material nonlinearity;

iv) Explicit and implicit integration loops;

v) Simplified I/O features;

vi) Accessible code architecture;

vii) Extensively benchmarked; and

viii) Requisite portability.

In terms of the squeeze film damper element noted in Report 1 [7], extensive modifications were introduced into the ADINA code. These modifications were made general enough so as to handle rotor-bearing-stator simulations involving any number of rotors and associated squeeze film damper elements. Recalling Chapter 4, the program has two available solution procedural loops, namely either explicit [12] or implicit [13].

Since extensive amounts of data are generated during a typical run, graphics post processors have also been developed to simplify output evaluation. These include both 2-D as well as 3-D plotter schemes. The 2-D processor enables the plotting of:

i) Bearing/rotor trajectories at a given station;

ii) Stator trajectories;

iii) Clearance histories at a given bearing station; as well as,

iv) Force, velocity and acceleration histories at given bearing stations.

The 3-D processor enables the plotting of isometric views of the rotor trajectories.

For convenience, Appendicies 2-5 give detailed discussions and listings of the ADINA modifications and graphics post processors. Included

are extensive instructions on user I/O of ~~both the~~ ADINA Bearing Code and the associated graphics routines. Specifically:

i) Appendix 2: I/O instruction on modified ADINA code enabling rotor-bearing-stator simulations;

ii) Appendix 3: Code listing of ADINA modifications;

iii) Appendix 4: Code listing and I/O instructions for 2-D plot routine;

iv) Appendix 5: Code listing and I/O instructions for 3-D plot routine.

Note the bearing modifications incorporated into ADINA were developed so as not to disturb the I/O flow. Hence, apart from required bearing data (film properties, clearances, etc.), the standard ADINA I/O remains intact [8]. For specific details, see Appendicies 2-5.

6. NUMERICAL OPERATIONAL CHARACTERISTICS

As noted earlier, this section will determine the numerical operational characteristics of the direct integration approach of FE generated rotor-bearing-stator simulations. This will include such items as:

a) Benchmarking;

b) Comparison of explicit vs. implicit methodologies of direct integration; and

c) Demonstration problems.

6.1 Benchmarking

To benchmark the overall procedure, a simple lumped parameter direct integration scheme was developed. [7] This approach was used to check the accuracy of the FE generated scheme involving the ADINA "implant". As the first example of such benchmarking, consider the system defined in Fig. (6.1). The material and geometric properties associated with the dual ported squeeze film damper bearing employed in this and the following sample problems are defined by

Nominal diameter = 6 inches; nominal length = 1.2 inches;
clearance = 1. inch; viscosity = $.1*10^{-5}$;
film rupture pressure = 15 psia;
$\theta_1 = 90°$; $\theta_2 = 270°$; $P_1 = 15$ psi; $P_2 = 55$ psi.

Note as with current practice, the bearing used in the simulation has no centering spring. In terms of this system, Figs. (6.2-6.10) illustrate various aspects of the validation. Specifically, the sets of results given by Figs. (6.2-6.4), (6.5-6.7), and (6.8-6.10) define checks involving respectively the Newmark [18], Wilson [17], and central difference [16]

operators. In this way, both the implicit and explicit schemes are treated. In each of the comparisons, three different aspects of rotor-bearing-stator behavior are depicted specifically:

a) Rotor displacement trajectories;

b) Stator displacement trajectories; and,

c) Relative rotor orbit.

For the present purposes, while extensive benchmarking was undertaken, for convenience, only the case of mild load imbalance is depicted. Note, due to the complexity of the various trajectories, comparisons between the FE and benchmarking schemes [7] are given soley for the relative rotor orbits. These are shown in Figures (6.4), (6.7), and (6.10). As can be seen, for the given Δt steps chosen, good comparisons were obtained by both the implicit and explicit schemes. Such benchmarking was obtained over a wide range of rotor speeds and load imbalance levels. In all cases, good accuracy was yielded.

Similar benchmarking was also performed for multibearing-rotor-stator simulations. Note, so long as Δt was kept small, good accuracy was obtained over a wide range of system variables.

6.2 Comparison of Explicit - vs. - Implicit Methodologies of Direct Integration

For problems involving few degrees of freedom, it was found that for a given accuracy, both the implicit and explicit schemes required about the same overall computational times. This follows from the fact that for "small problems" the architectural overhead associated with the implant strategy programming dominates over the relative algorithmic efficiency. Note for transients initiated by rather severe imbalance loads, it was found that the implicit scheme proved to be more sensitive to the

choice of time step size. Interestingly, such sensitivities were found to occur for problems with small as well as large numbers of degrees of freedom. After performing several parametric studies, it was found that during the course of a typical transient, particularly involving a severe loading, the tangent properties of the fluid film undergo major changes. Because of this, a perfusion of system harmonics are introduced into the transient response as the tangent fluid film properties vary. Note, this behavior is intrinsic to the fluid film and hence, is independent of the number of degrees of freedom of the rotor-stator model. Such properties tend to reduce the stability threshold of the implicit scheme which is best employed when only a few harmonics are excited. For problems involving large numbers of degrees of freedom, the sensitivity of the implicit scheme coupled with the required continuous updating and inversions of the dynamic stiffness tends to reduce the running efficiency of the procedure.

In contrast, the central difference approach tends to be less sensitive to such tangent property fluctuations. This is true so long as the resulting family of harmonics is bound by the spectral characteristics of the rotor-stator system. Because of this, similar Δt can be employed by the implicit and explicit scheme. In this context, the explicit scheme is somewhat more computationally efficient than the implicit approach. This follows from the fact that no continuous inversion is required by the explicit scheme.

6.3 Demonstration Problems

To demonstrate various of the capacities of the ADINA implant strategy, the results of several example problems will be considered. This will involve single and multiple bearing problems with various types of load

imbalance histories, impact events, and rotor speeds. For example, in terms of the single bearing system given in Fig. (6.1), Figs. (6.11 - 6.13), illustrate various aspects of the response histories to an imbalance load which is applied as a ramp function in time. As this loading is more severe than that applied in Figs. (6.2 - 6.10), the rotor tends to fill its clearance circle as seen in Fig. (6.13).

If the same system is subject to a undirectional impulse, as might be expected from a rough landing, Figs. (6.14 - 6.16) illustrate the associated response history. By comparing Figs. (6.11, 6.12) with (6.14, 6.15), the effects of the impulse can be clearly seen from the ovalizing of the rotor and stator trajectories. This is directly due to the directional characteristics of the impulse load. Note, comparing Fig. (6.13) with (6.16), we see that the rotor orbits are essentially unchanged by the presents of the shock load. Hence, it follows that the squeeze film damper has essentially no effect on mitigating the worst aspects of unidirectional shocks when large load imbalances are involved.

For the current problem this is a direct outgrowth of the fact that when large orbits occur, the oil film being thin tends to have very high tangent stiffness properties. Because of this, the squeeze film damper does little to filter or dampen the exciting load. In this context, such damper bearing systems will have only a small effect on mitigating the severity of maneuvering and landing loads on rough running engines.

The next series of examples pertains to the rotor-bearing-stator system defined in Fig. (6.17). For this system only one squeeze film damper is employed. To simulate the shafting, the beam elements available in the ADINA system are utilized. Note, the mass effects are handled via the lumped paramete approach. The series of figures given by (6.18 - 6.20), (6.21 - 6.23) and (6.24 - 6.26) respectively illustrate the effects

of increasing the severity of loading on the dynamic response of multi-bearing problems. As can be seen by examining the progression of figures defining the rotor/stator displacements and the rotor orbit, increasingly stiffer squeeze film damper responses are excited. This is clearly seen from the wide rotor orbits which occur. Note, as the load is further increased, the rotor stator trajectories become "locked in".

The next series of figures namely (6.27 - 6.29), illustrate the effects of suddenly applied load imbalances. Since the rotor speed considered is high, only a small portion of the clearance circle is filled. This follows from the fact that due to the rather rapid changes in orientation of the exciting load, the fluid does not have sufficient time to recirculate. Because of this the rotor is supported by severe velocity gradients which are only in a close neighborhood. Hence, the pressure gradient generated by the inlet and outlet ports of the squeeze film device causes the rotor to settle in the direction of the low pressure port. Note, as the rotor speed is decreased, increasingly larger rotor orbits occur. Similar trends occur as the level of imbalance is increased.

As a last example, consider the system defined in Fig. 6.30. Based on this configuration, Figs. (6.31 - 6.36) depict various aspects of the response of the system configured with two squeeze film dampers. Similar to the previous problems, the exciting forces involve imbalance loads which are applied in a ramp fashion in time. As can be seen from the results depicted for the lightly and heavily loaded squeeze film damper bearings, the orbits are respectively small and large.

Note, based on numerous parametric studies involving systems similar to the foregoing, it was found that all the time integration schemes considered were stable for situations wherein the fluid underwent only moderate

changes in stiffness during the overall cycle. Note, while a perfusion of harmonics is introduced by even moderate changes in stiffness, so long as the resulting spectra are strongly bound by the frequency envelop of the dominate system frequencies, spurious energy flow to higher order modes is insignficant. Specifically, for the implicit scheme, if the choice of time step size is gauged to the dominate higher order system frequencies, then the introduction of lower order spectra by the squeeze film has little effect on numerical stability. In contrast, if strong stiffness modulations occur, then significant amounts of energy flow are introduced in the ever shifting higher order modes. This leads to solution instabilities unless smaller Δt are introduced. In this context, as noted earlier, since the explicit approach tends to be more efficient for a given Δt, for problems involving strong vs. weak imbalance loads, its use is advocated over the implicit scheme.

7. SUMMARY

The main thrust of this report has been to outline the second year results of the Engine Dynamic Analysis with General Nonlinear Finite Element Codes Grant. This has included two main areas namely:

i) Implanting the squeeze film damper element into a general pur ose FE code for testing and evaluation and;

ii) Determining the numerical characteristics of the FE generated rotor-bearing-stator simulation scheme.

Due to the generality of the approach taken, the overall implant strategy and associated algorithmic schemes has several important features namely:

i) Handles hierarchal rotor-bearing-stator system involving the possibility of series and parallel substructure; hence, series and spoolel engine configurations can be handled;

ii) Handles large deformation kinematics and kinetics;

iii) Implant architecture is well suited to handle the possibility of rubbing and contact;

iv) Due to the use of the ADINA code, a wide assortment of element and material types can be used in a simulation;

v) To enable the incorporation of new time stepping algorithms, both an explicit and implicit formulation has been incorporated in the implant and;

vi) 2-D and 3-D graphics plotter schemes have been developed to enable the post processing of response data; this includes;

(a) Bearing/Rotor trajectories;

(b) Stator trajectory;

(c) Rotor Orbit;

(d) Force velocity and acceleration histories at a given location and finally;

(e) The 3-D processor enables the plotting of isometric views of the rotor shape at anytime.

8. ACKNOWLEDGEMENT

The authors gratefully acknowledge the many fruitful discussions with Chris Chamis of NASA Lewis which made this effort possible.

9. REFERENCES

1. Noor, A., "Survey of Computer Programs for Solution of Nonlinear Structural and Solid Mechanics Programs," Comp. Struct. 13, 425-465 (1981).

2. Chang, T. Y. and Padovan, J., "General Purpose Finite Element Programs," Structural Mechanics Software Series, Vol. III, edited by Perrone, N. and Pilkey, W., University Press of Virginia, Charlottesville (1980).

3. Zienkiewicz, O. C., "The Finite Element Method," McGraw-Hill Book Co., New York (1977).

4. Bathe, K. J. and Wilson, E. I., "Numerical Methods in Finite Element Analysis," Prentice Hall, Englewood Cliffs, New Jersey (1976).

5. Adams, M. L., "Nonlinear Dynamics of Multi-Bearing Flexible Rotors," Jr. Sound and Vibration, 71, 129-144 (1980).

6. Guyan, R. J., "Reduction of Stiffness and Mass Matrices," AIAA Jr., 3, 380 (1965).

7. Adams, M. L., Padovan, J. and Fertis, D. G., "Engine Dynamic Analysis with General Nonlinear Finite Element Codes, Part 1: Overall Approach and Development of Bearing Damper Element," ASME Paper No. 81-FT-151. Also, "Finite Element for Rotor/Stator Interactive Forces in General Engine Dynamic Simulation, Part 1: Development of Bearing Damper Element," NASA CR-165214 (1981).

8. Bathe, K. J., "ADINA - A Finite Program for Automatic Dynamic Incremental Nonlinear Analysis," Report 82448-1, Acoustics Lab, Mechanical Engineering Dept., MIT, May 1976 (revised 1977).

9. Padovan, J., Chang, T. Y., "Evaluation of ADINA: Part II, Operating Characteristics," Report No. AUE 802, University of Akron, prepared for ONR under Contract N0014-78-C-0691, (1980).

10. Padovan, J., "Self-Adaptive Incremental Newton-Raphson Algorithms," Symposium on Computational Methods in Nonlinear Structural and Solid Mechanics, Washington D.C., also NASA CP-2147 (1980).

11. Padovan, J., Tovichakchaikul, S., "Self-Adaptive Predictor-Corrector Algorithms for Static Nonlinear Structural Analysis," Comp. and Struct. (in press), also NASA CR-165410 (1981).

12. Belytschko, T., "A Survey of Numerical Methods and Computer Programs for Dynamic Structural Analysis," Nucl. Eng. and Design, 37, 23-34 (1976).

13. Hughes, T. J. R., "Stability, Convergence and Growth and Decay of Energy of the Average Acceleration Method in Nonlinear Structural Dynamics," Jr. Comp. and Struct., 6, 313-324 (1976).

14. Fung, Y. C., "Foundations of Solid Mechanics," Prentice-Hall, Inc., Englewood Cliffs, New Jersey (1965).

15. Archer, J. S., "Consistent Mass Matrix for Distributed Mass Systems," Proc. Am. Soc. Civil Engrs., 89, 161-178 (1903).

16. Hildebrand, F. B., "Finite-Difference Equations and Simulations," Prentics-Hall, Inc., Englewood Cliffs, New Jersey (1968).

17. Wilson, E. L., "A Computer Program for the Dynamic Stress Analysis of Underground Structures," Report No. SESM68-1, Department of Civil Engineering, University of California, Berkeley (1968).

18. Newmark, N. M., "A Method of Computation for Structural Dynamics," J. Eng. Mech. Div., ASCE, 85, EM3, 67-94 (1959).

APPENDIX 1: FIGURES

ORIGINAL PAGE IS
OF POOR QUALITY

Figure 2.1 EXPLICIT SCHEME: FLOW OF CALCULATIONS

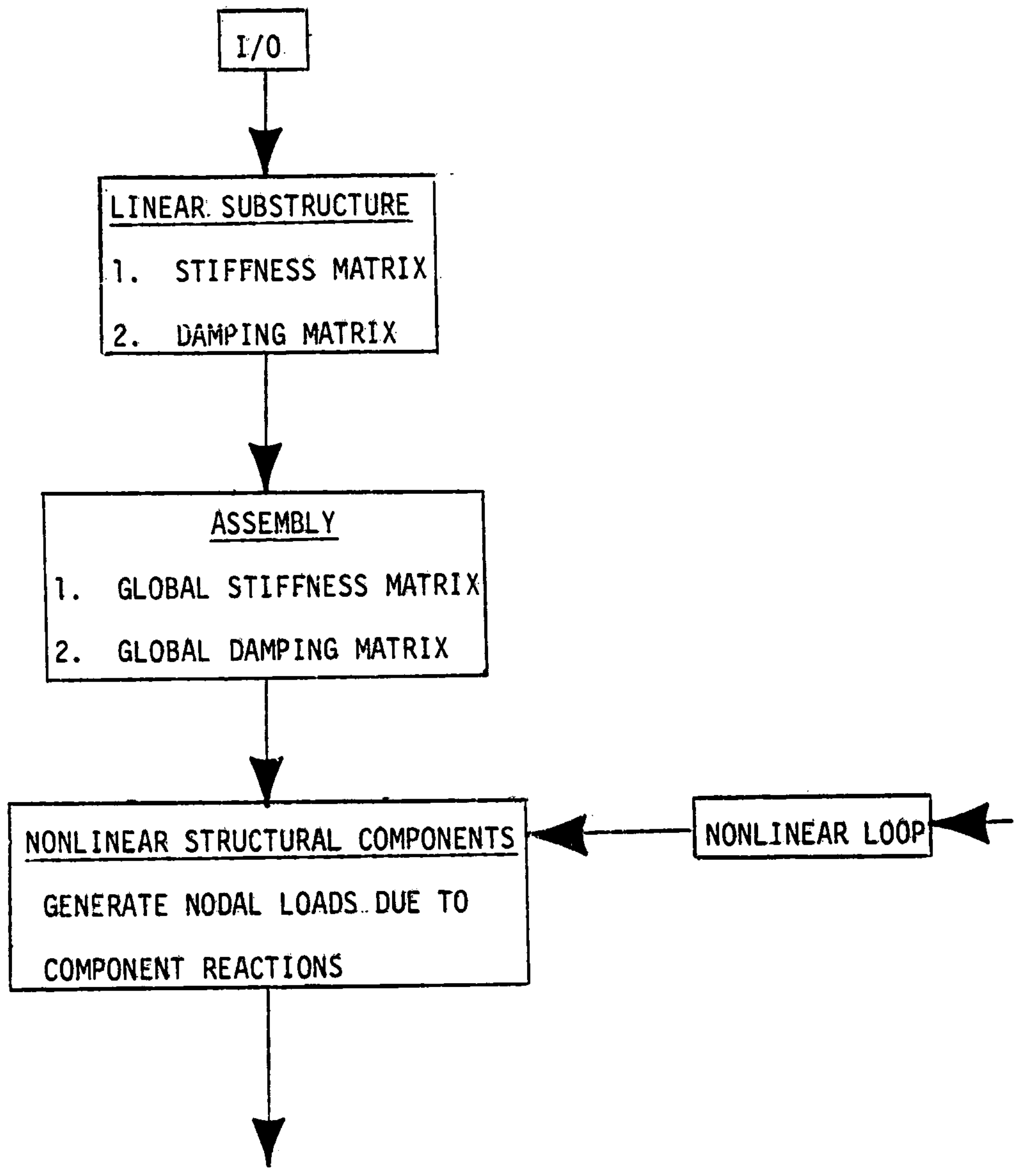
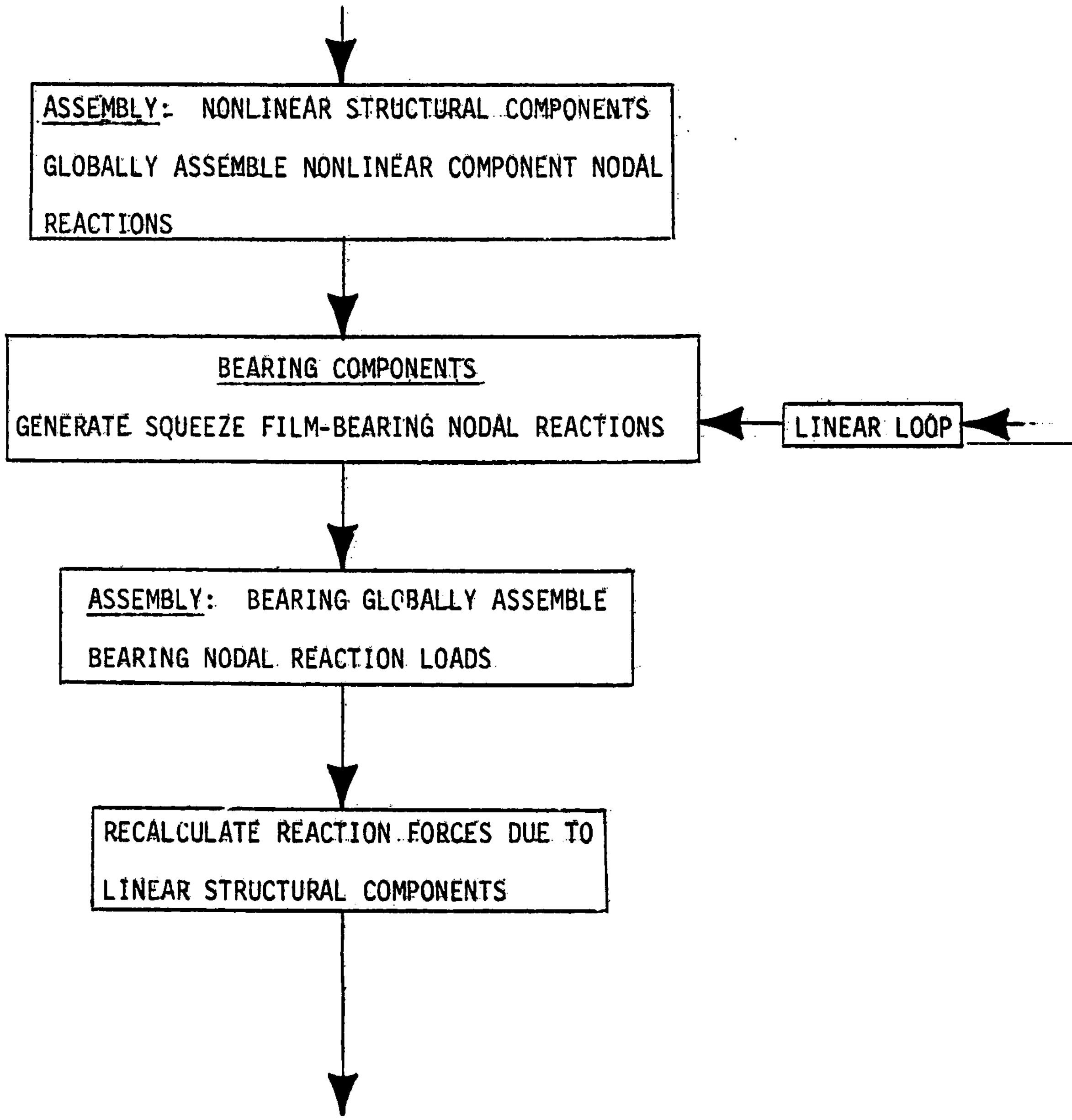


Figure 2.1 EXPLICIT SCHEME; FLOW OF CALCULATIONS (continued)



ORIGINAL PAGE IS
OF POOR QUALITY

Figure 2.1 EXPLICIT SCHEME; FLOW OF CALCULATIONS (continued)

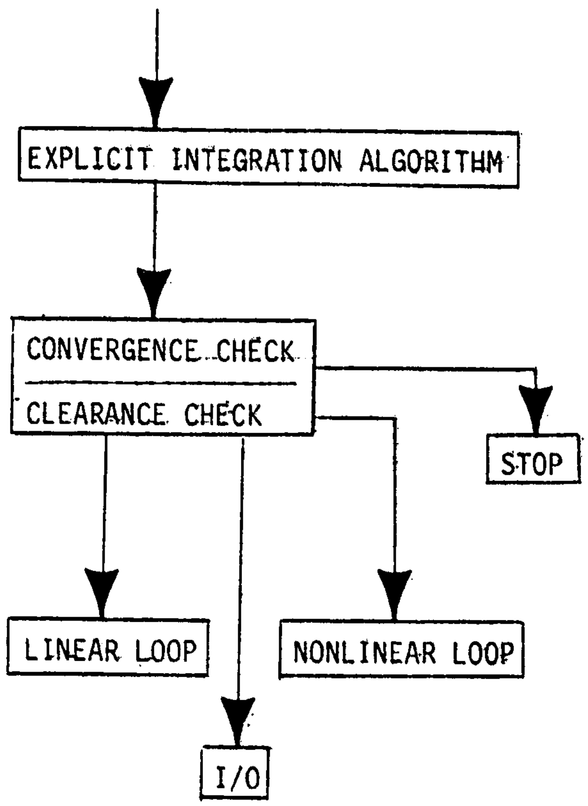


Figure 2.2 IMPLICIT SCHEME; FLOW OF CALCULATIONS

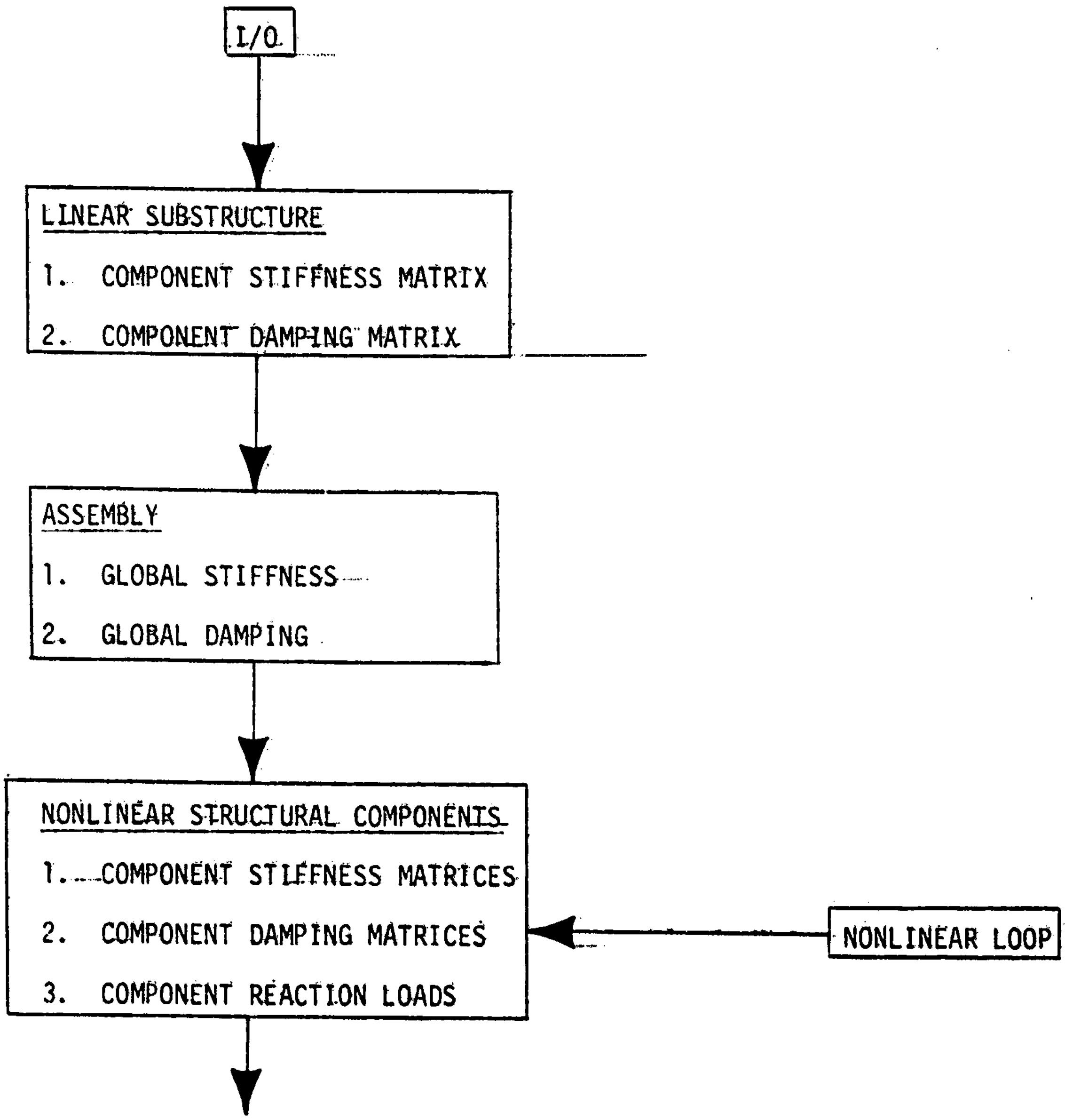
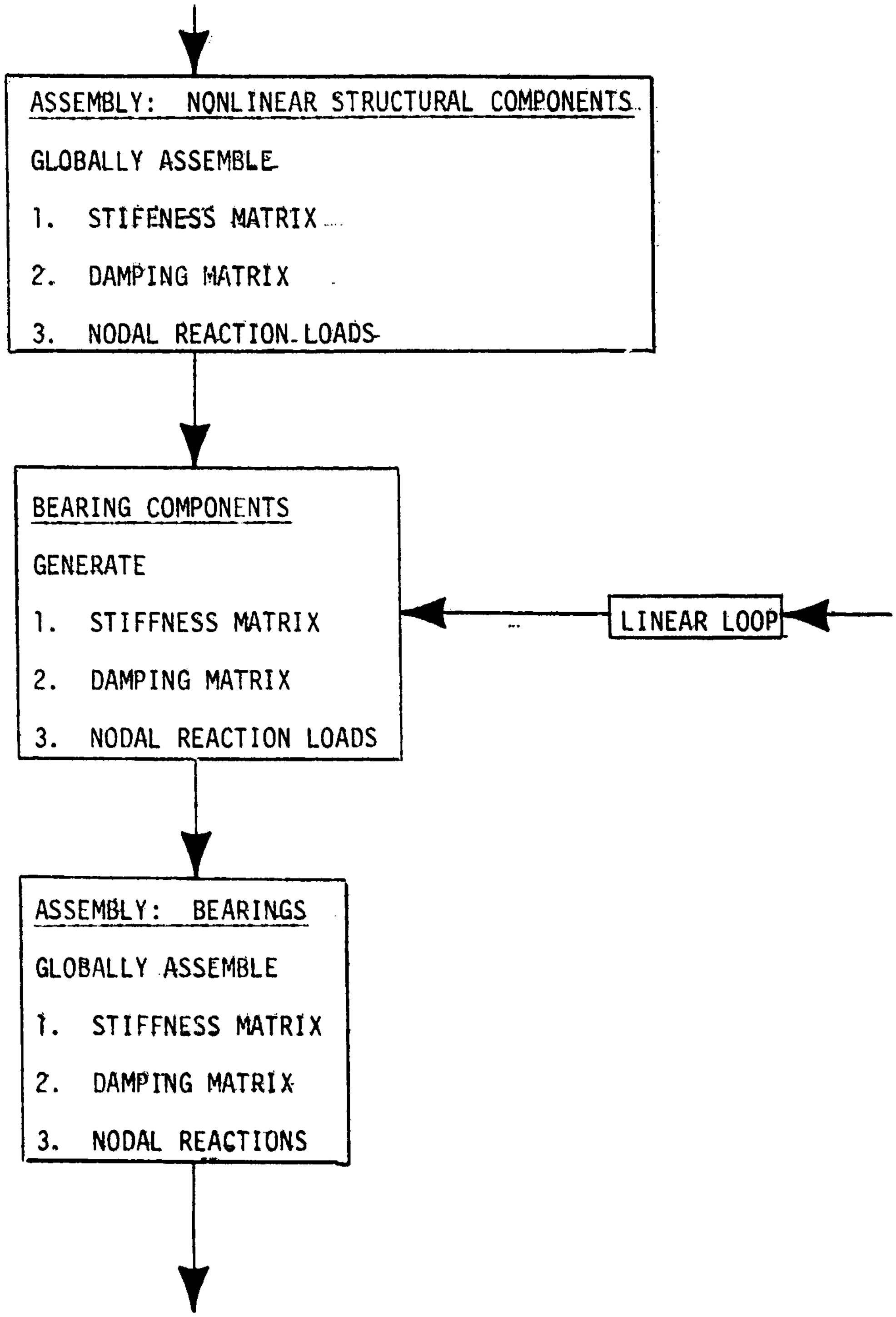


Figure 2.2 IMPLICIT SCHEME; FLOW OF CALCULATIONS (continued)



ORIGINAL PAGE IS
OF POOR QUALITY

Figure 2.2 IMPLICIT SCHEME; FLOW OF CALCULATIONS (continued)

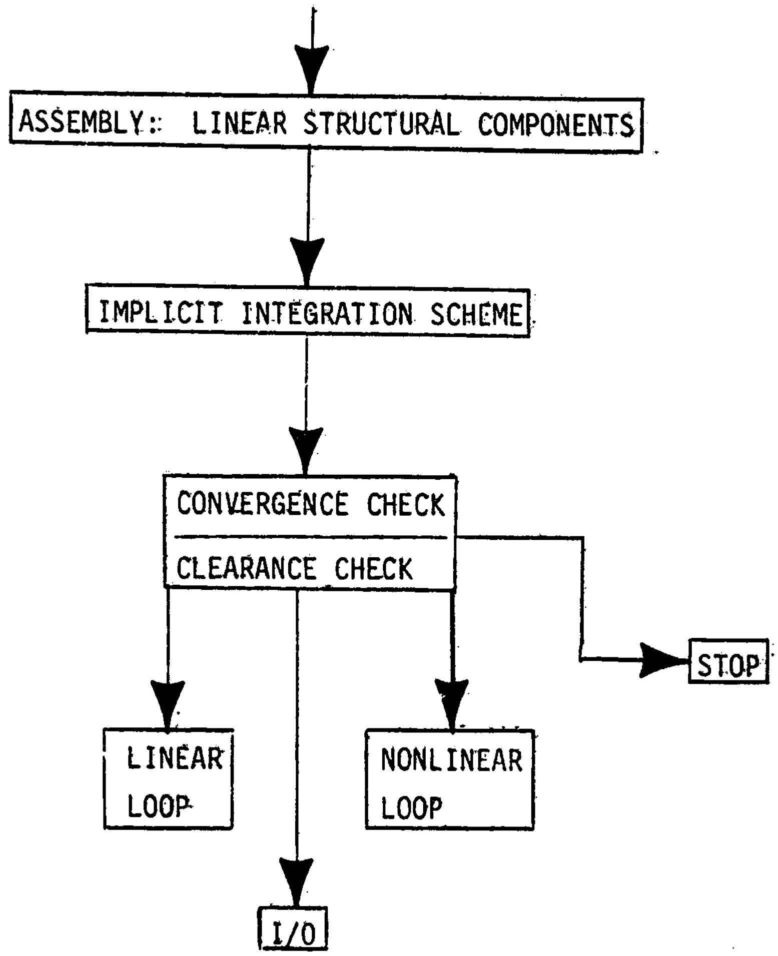


Figure 4.1 EXPLICIT SCHEME

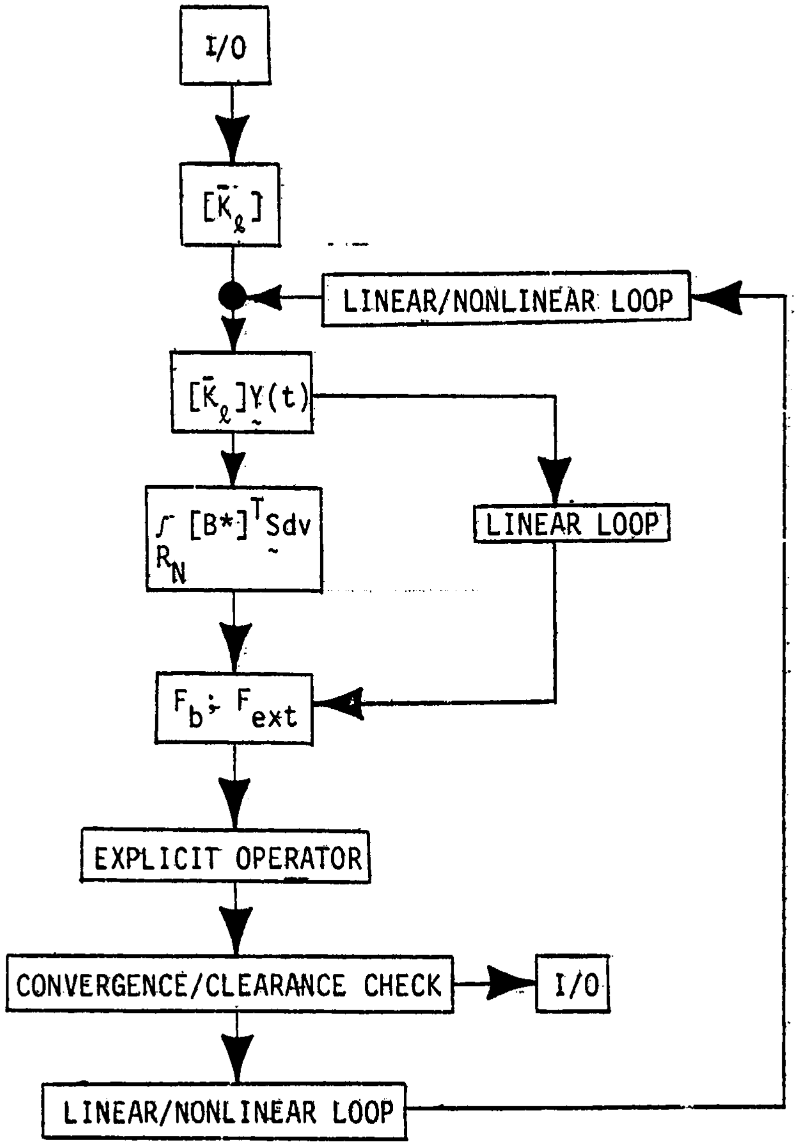
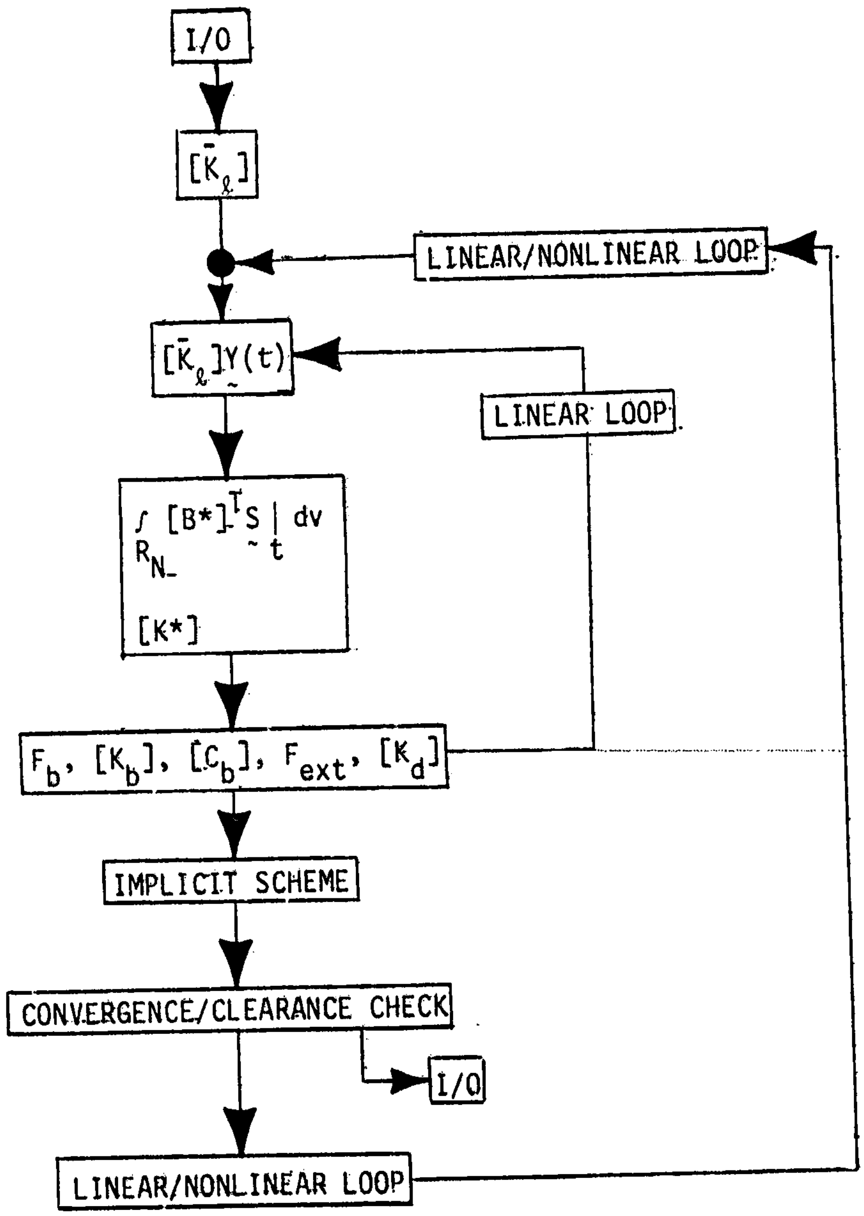


Figure 4.2 IMPLICIT SCHEME



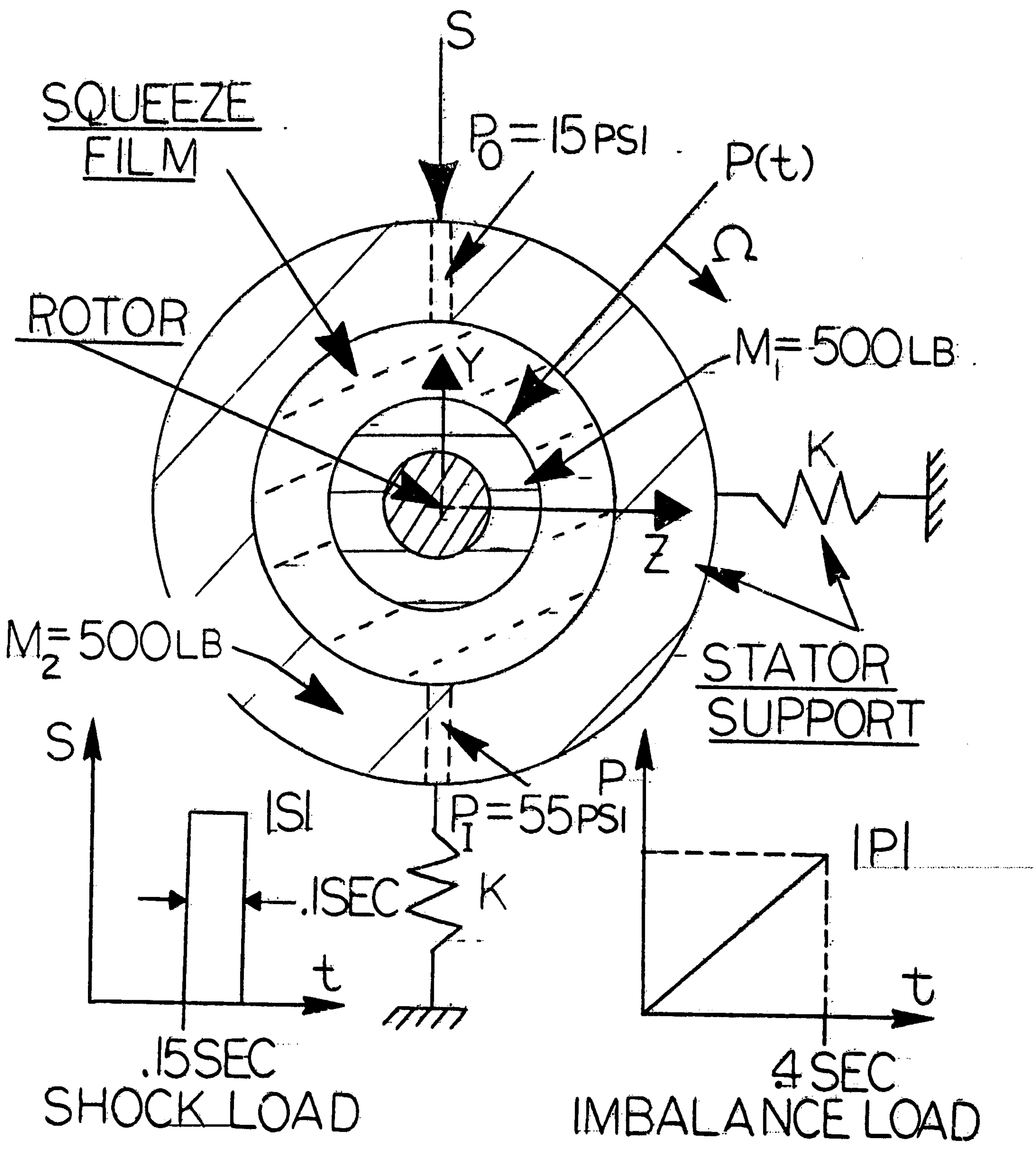


FIG.6.1 SINGLE BEARING MODEL

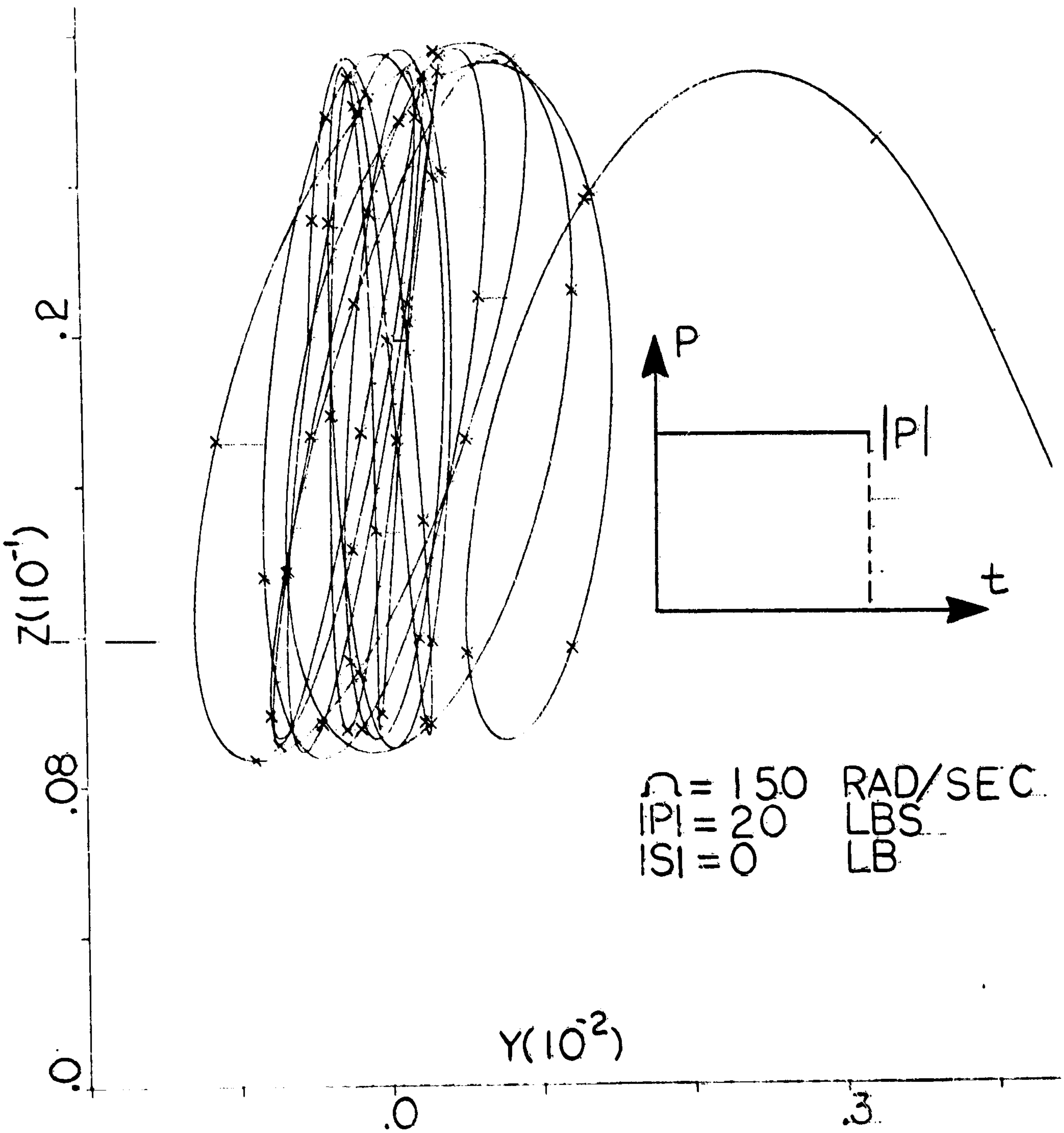


FIGURE 6.2 ROTOR DISPLACEMENT
NEWMARK METHOD
(SEE FIG. 6.1)

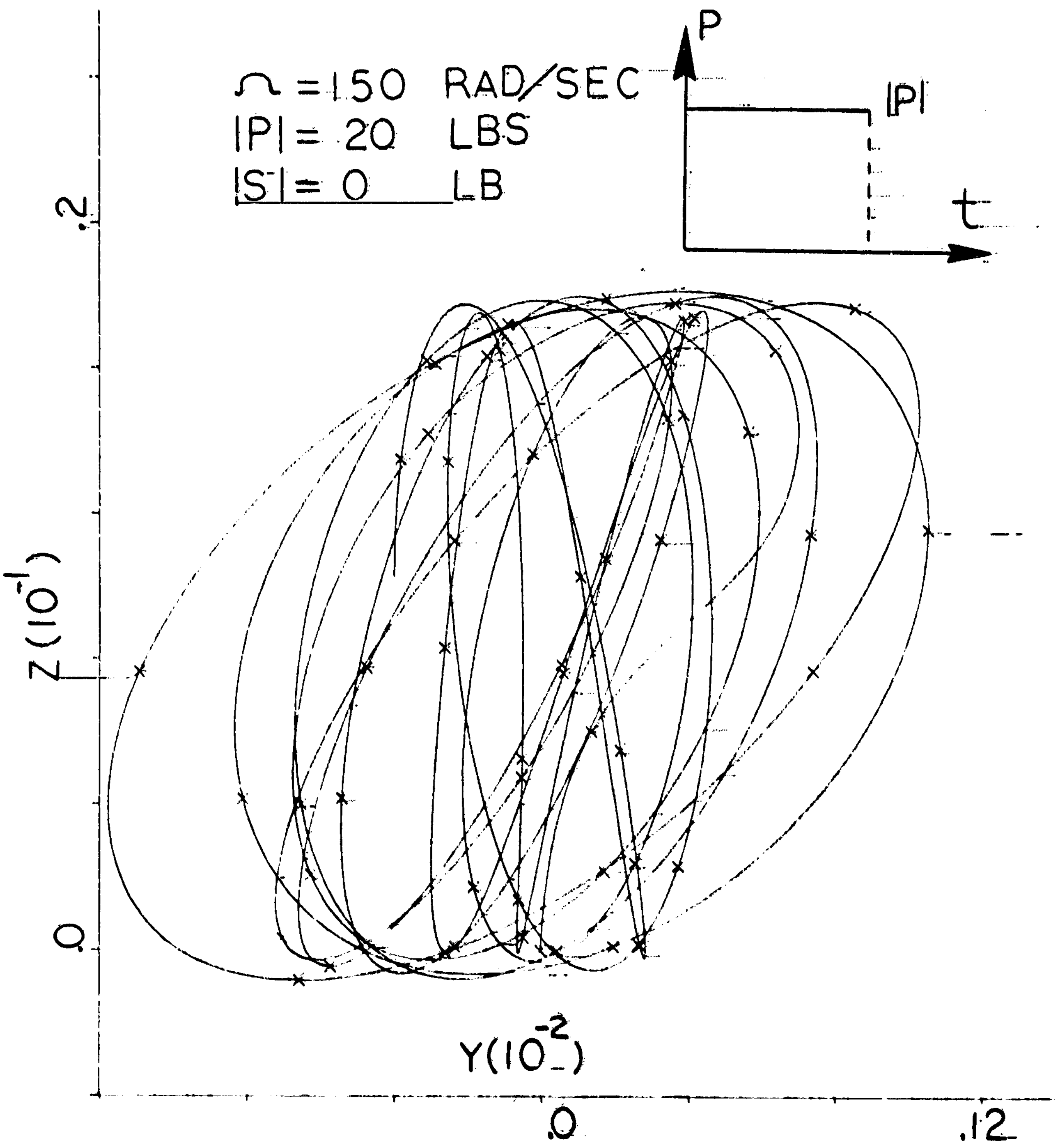


FIGURE 6.3 STATOR DISPLACEMENT
NEWMARK METHOD
(SEE FIG. 6.1)

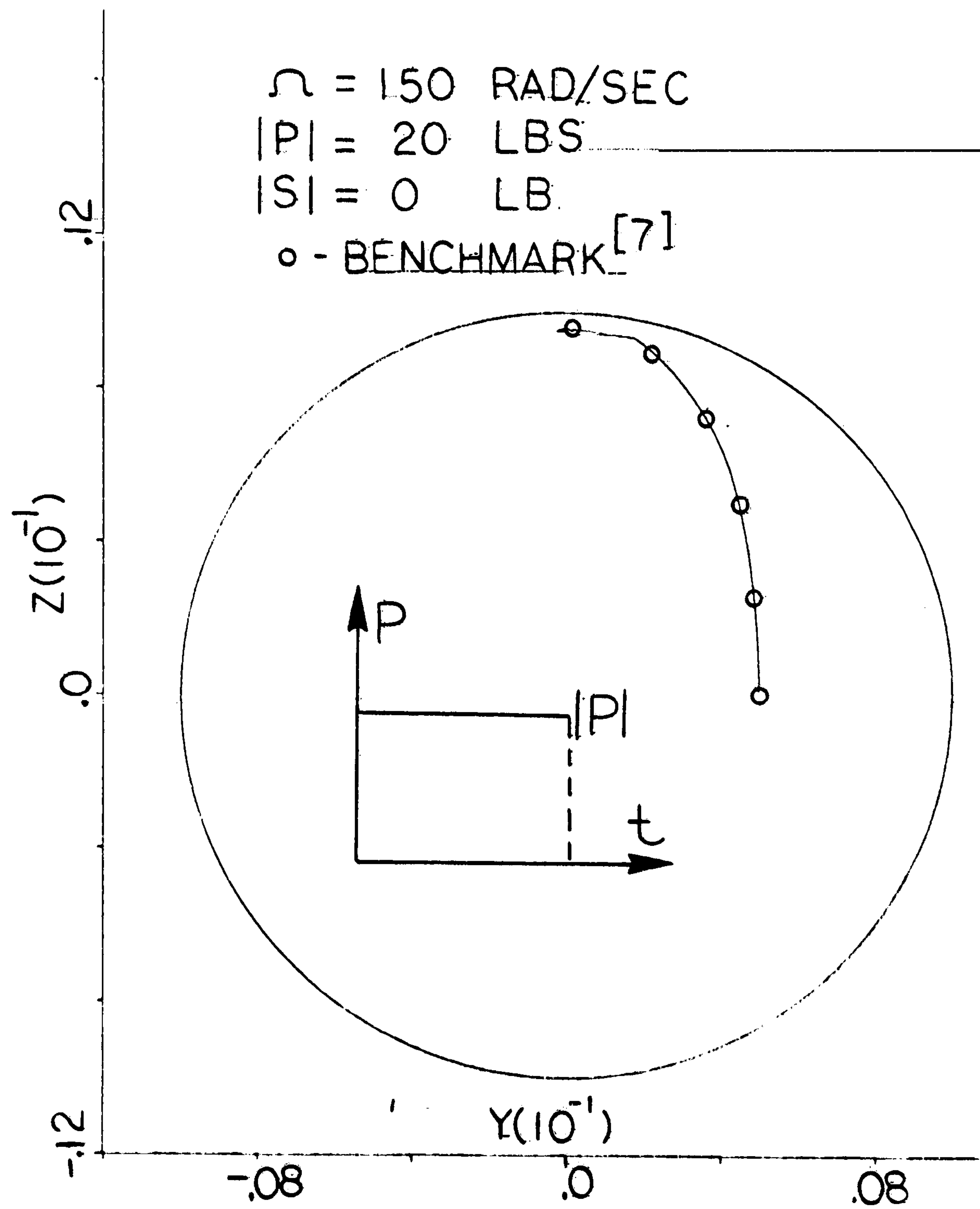


FIGURE 6.4 ROTOR ORBIT
NEWMARK METHOD
(SEE FIG. 6.1)

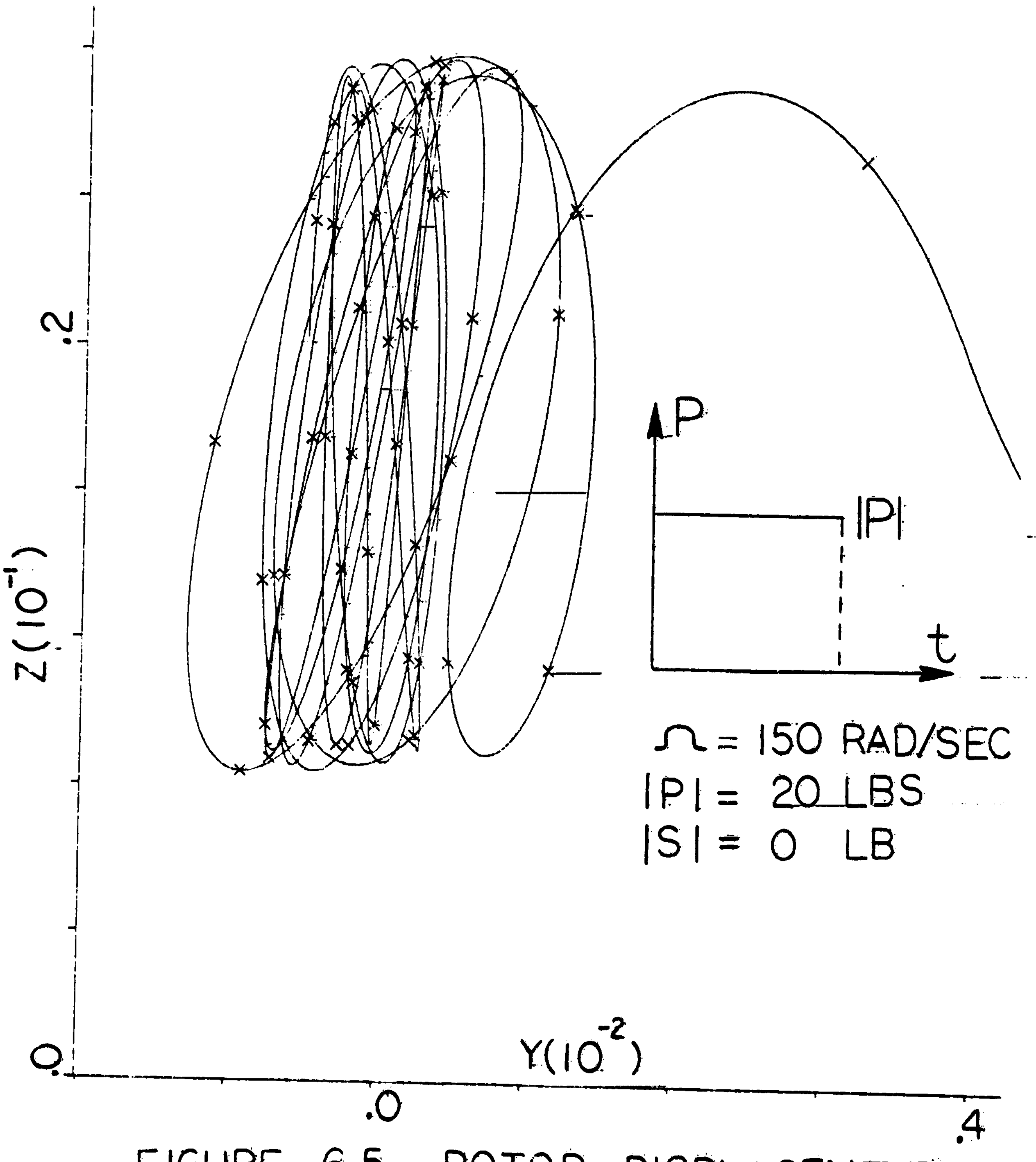


FIGURE 6.5 ROTOR DISPLACEMENT
WILSON METHOD
(SEE FIG. 6.1)

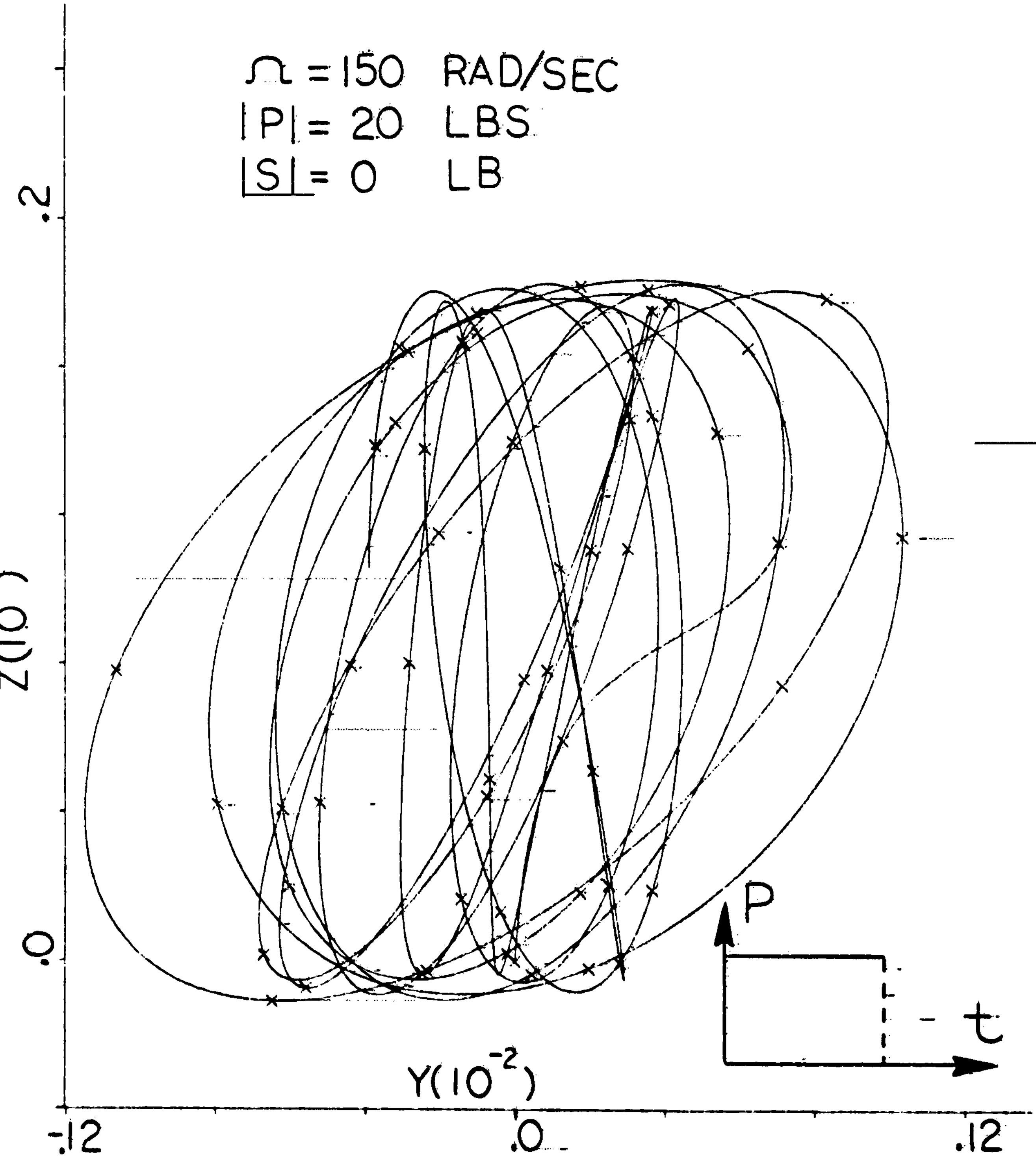


FIGURE 6.6 STATOR DISPLACEMENT WILSON METHOD (SEE FIG.6.1)

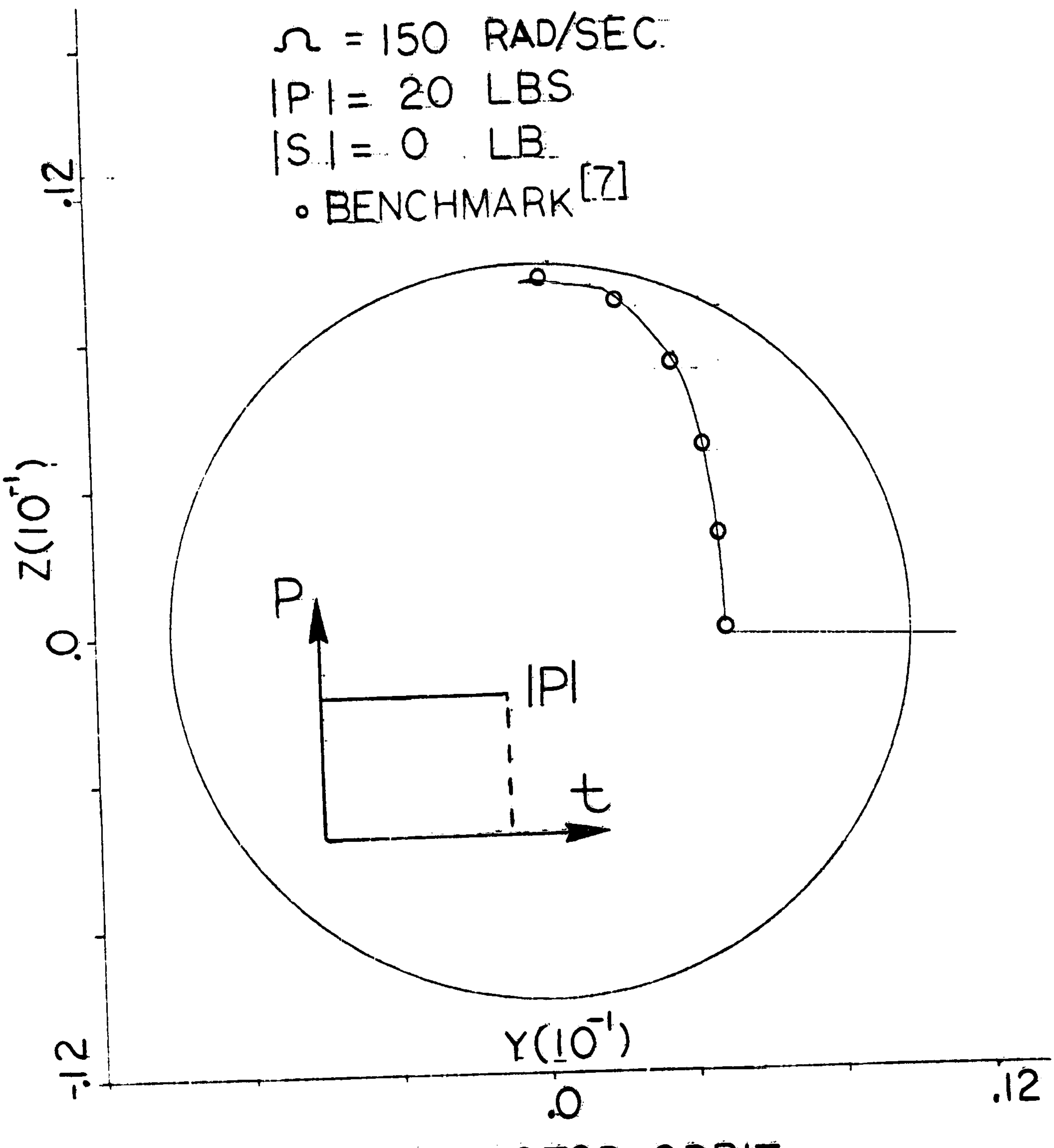


FIGURE 6.7 ROTOR ORBIT
WILSON METHOD
(SEE FIG. 6.1)

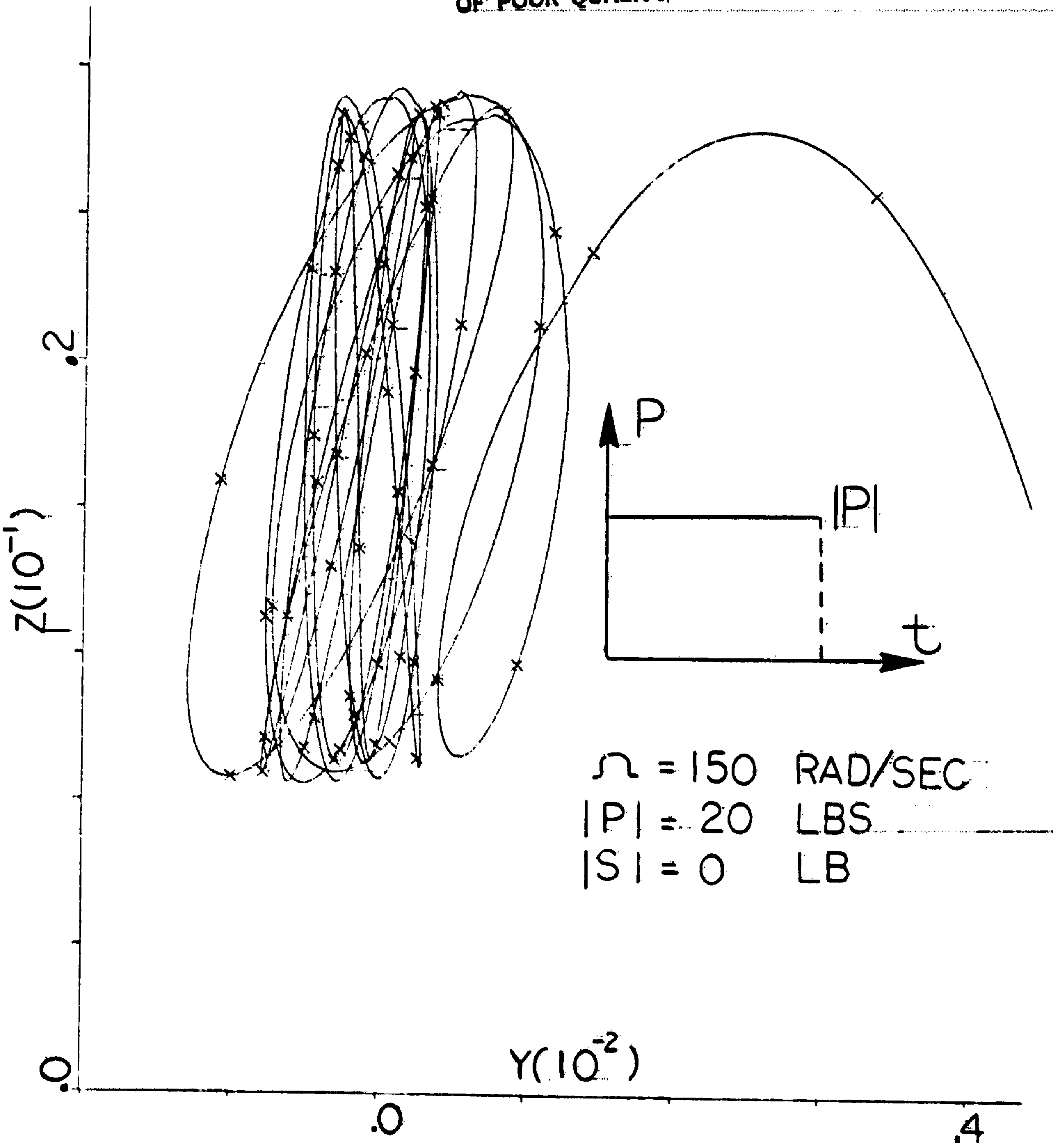


FIGURE 6.8 ROTOR DISPLACEMENT CENTRAL DIFFERENCE (SEE FIG. 6.1)

ORIGINAL PAGE IS
OF POOR QUALITY

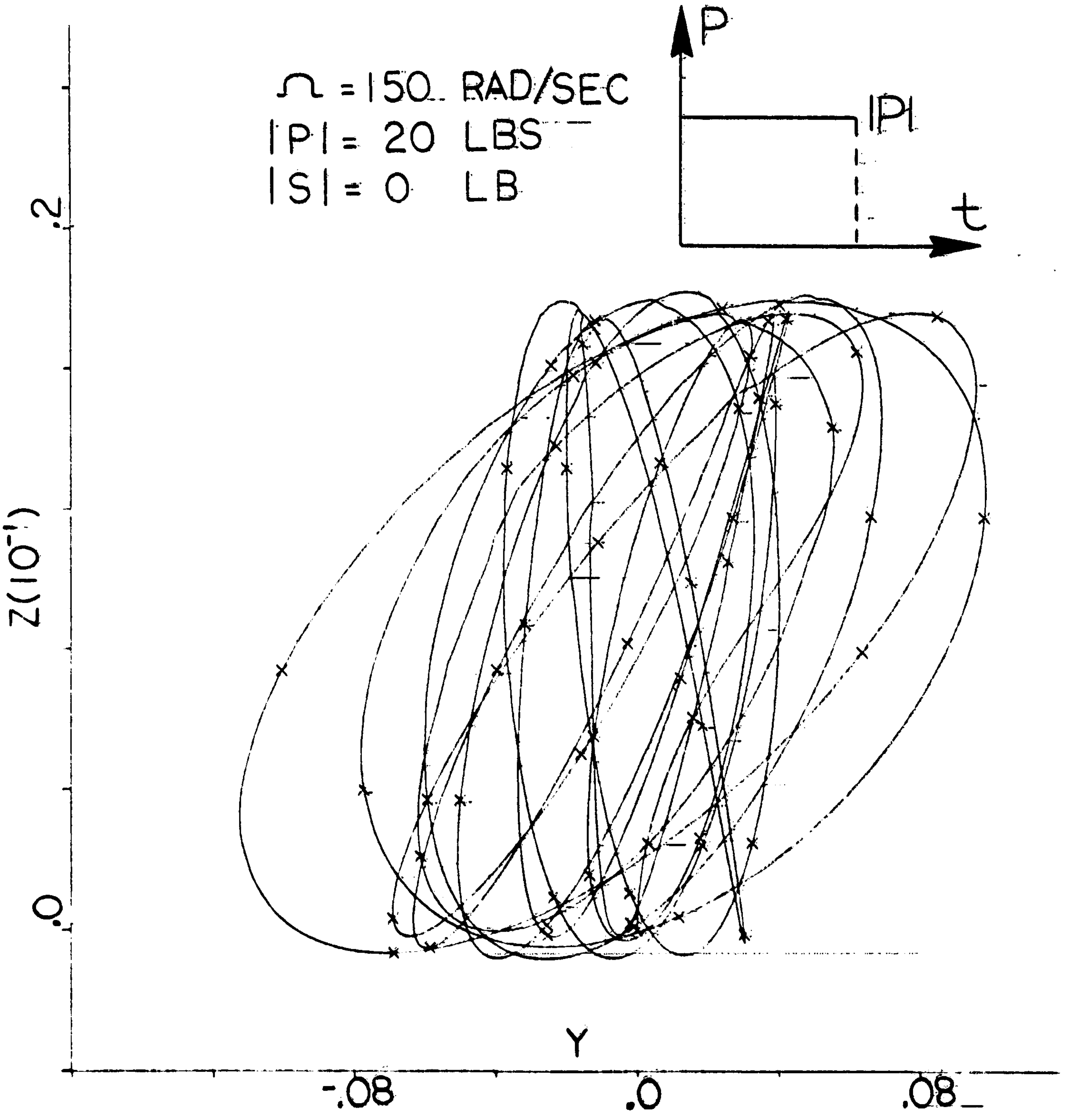


FIGURE 6.9 STATOR DISPLACEMENT
CENTRAL DIFFERENCE
(SEE FIG. 6.1)

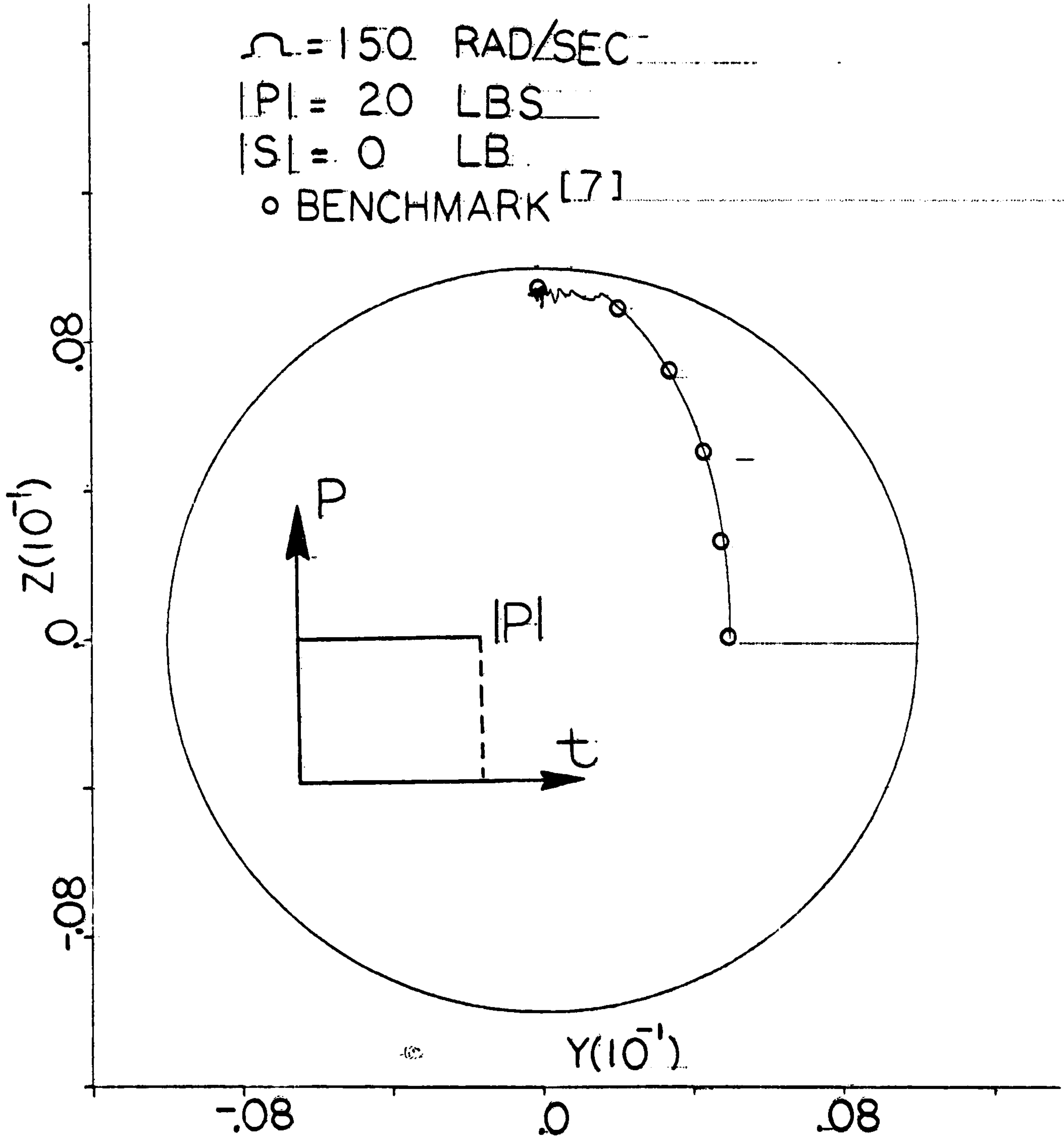


FIGURE 6.10 ROTOR ORBIT
CENTRAL DIFFERENCE
(SEE FIG. 6.1)

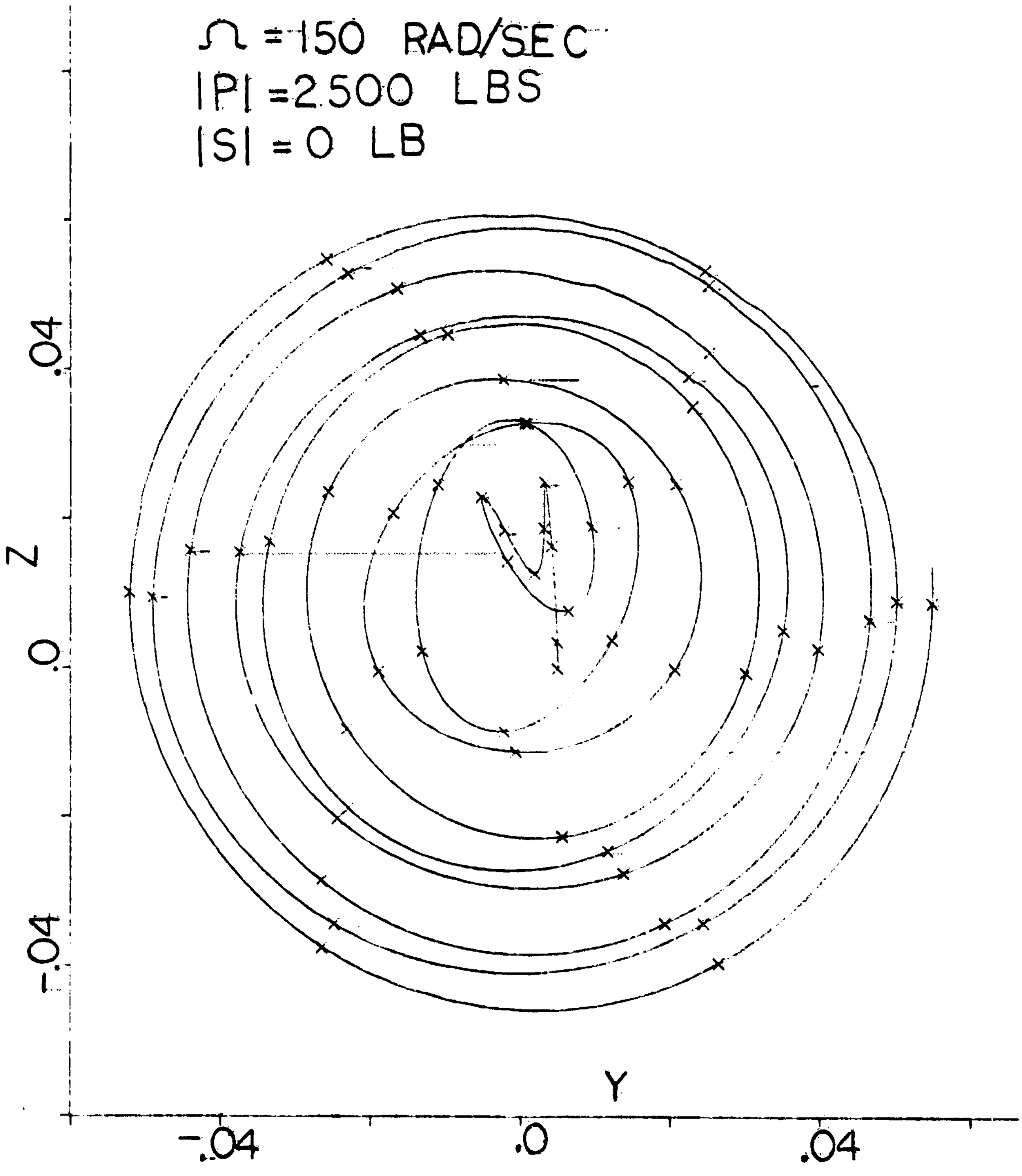


FIGURE 6.11 ROTOR DISPLACEMENT
CENTRAL DIFFERENCE
(SEE FIG. 6.1)

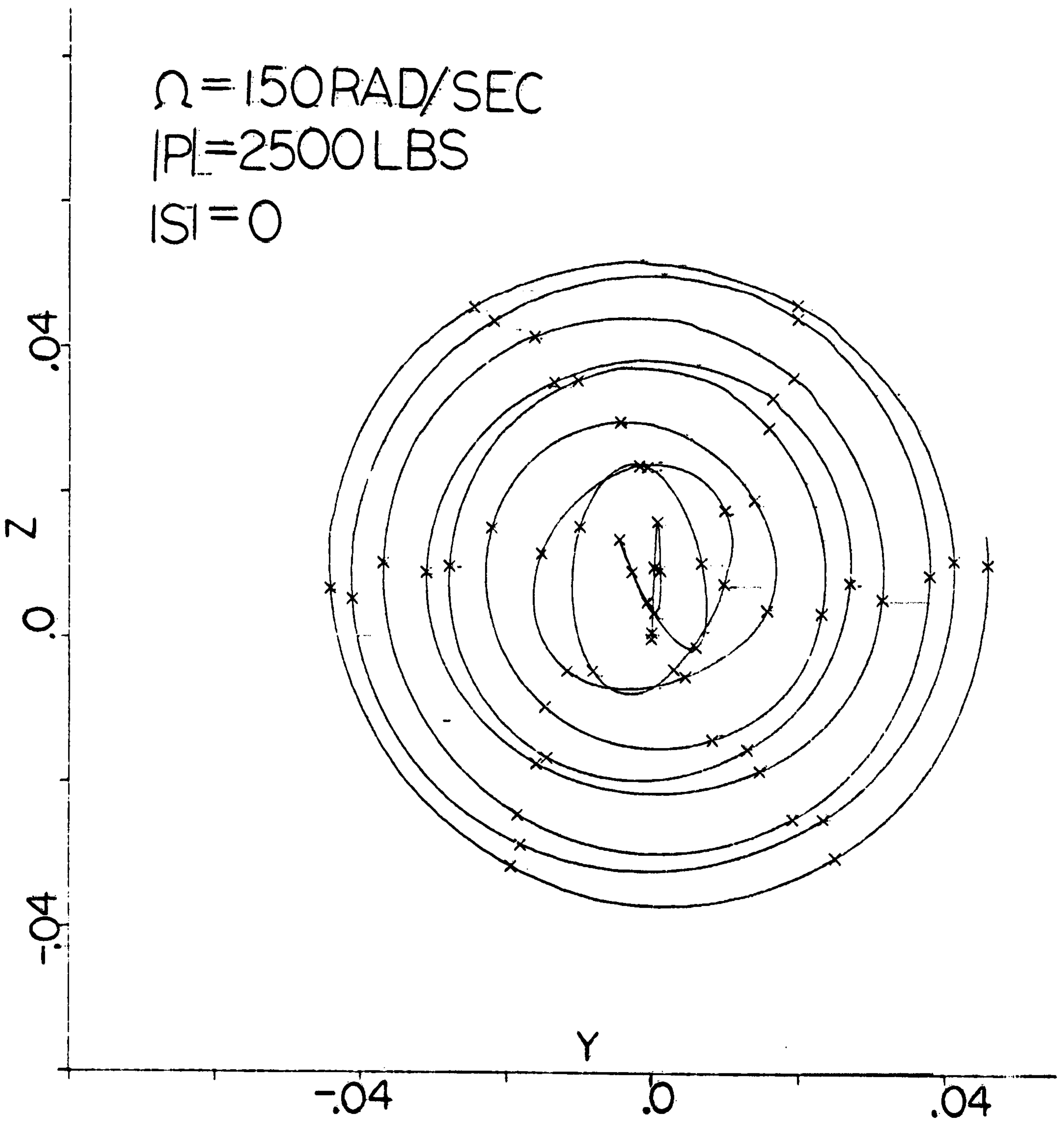


FIGURE 6·12 STATOR DISPLACEMENT
(SEE FIG. 6·1)

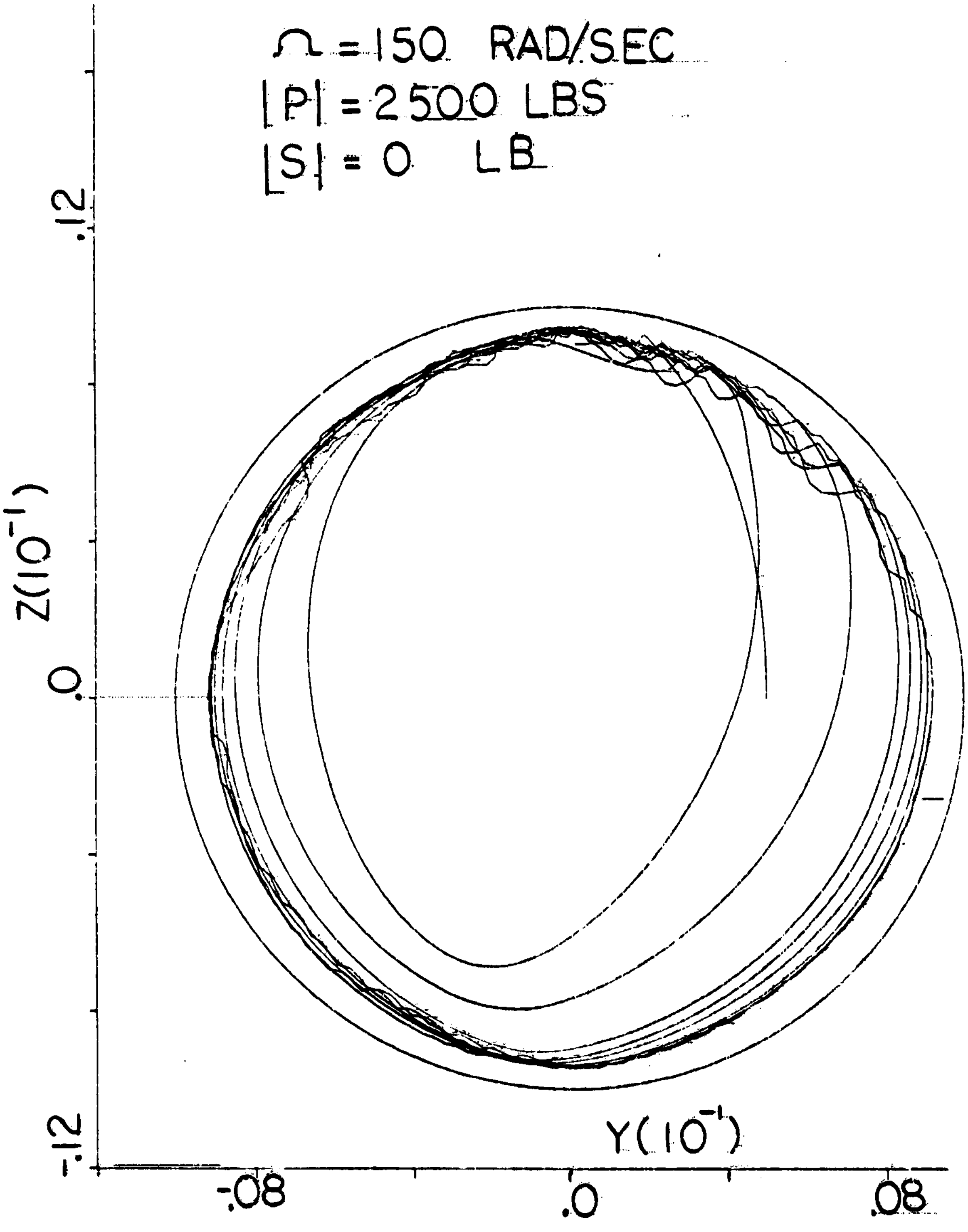


FIGURE 6.13 ROTOR ORBIT
(SEE FIG. 6.1)

ORIGINAL PAGE IS
OF POOR QUALITY

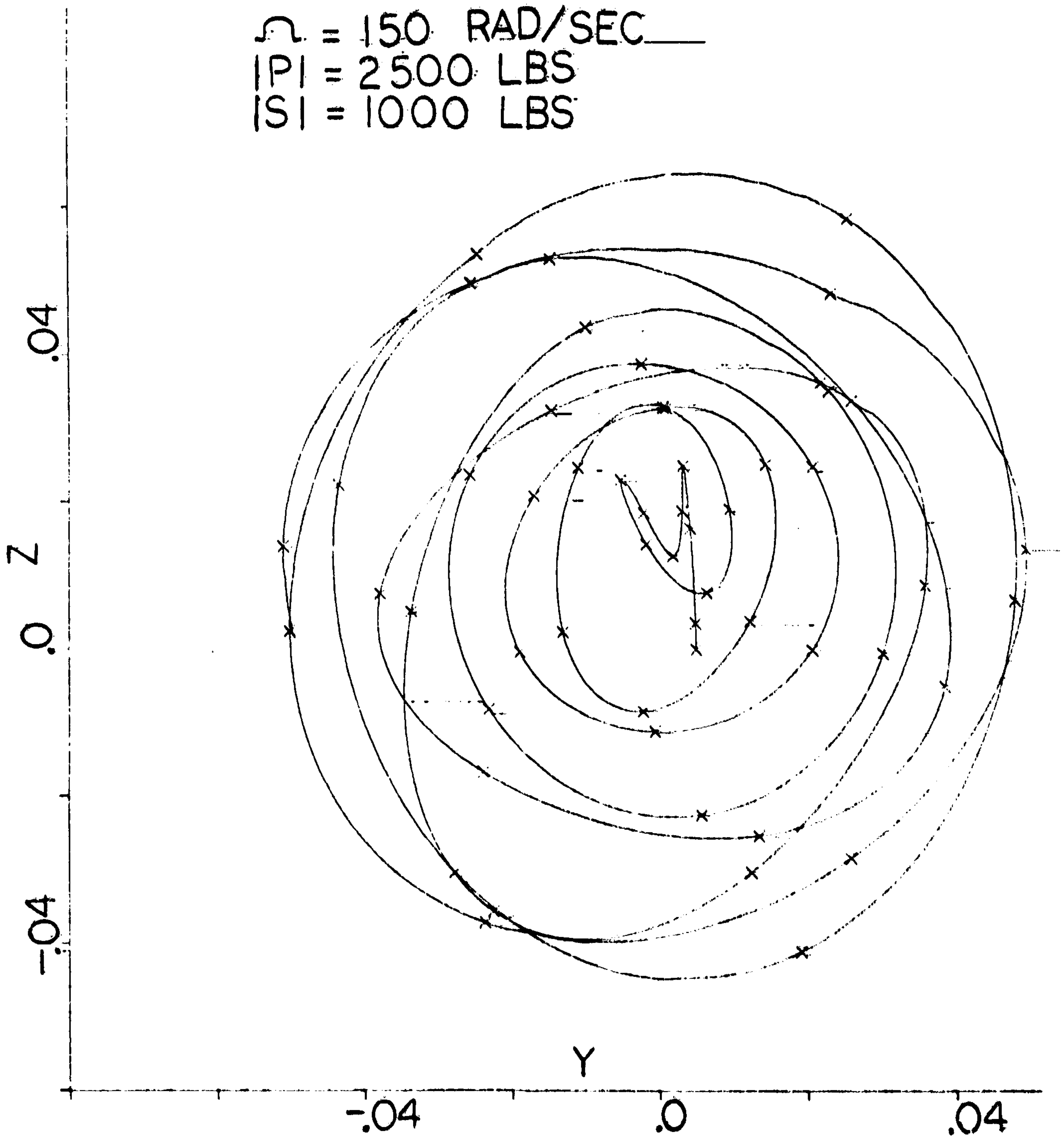


FIGURE 6.14 ROTOR DISPLACEMENT
(SEE FIG. 6.1)

ORIGINAL PAGE IS
OF POOR QUALITY

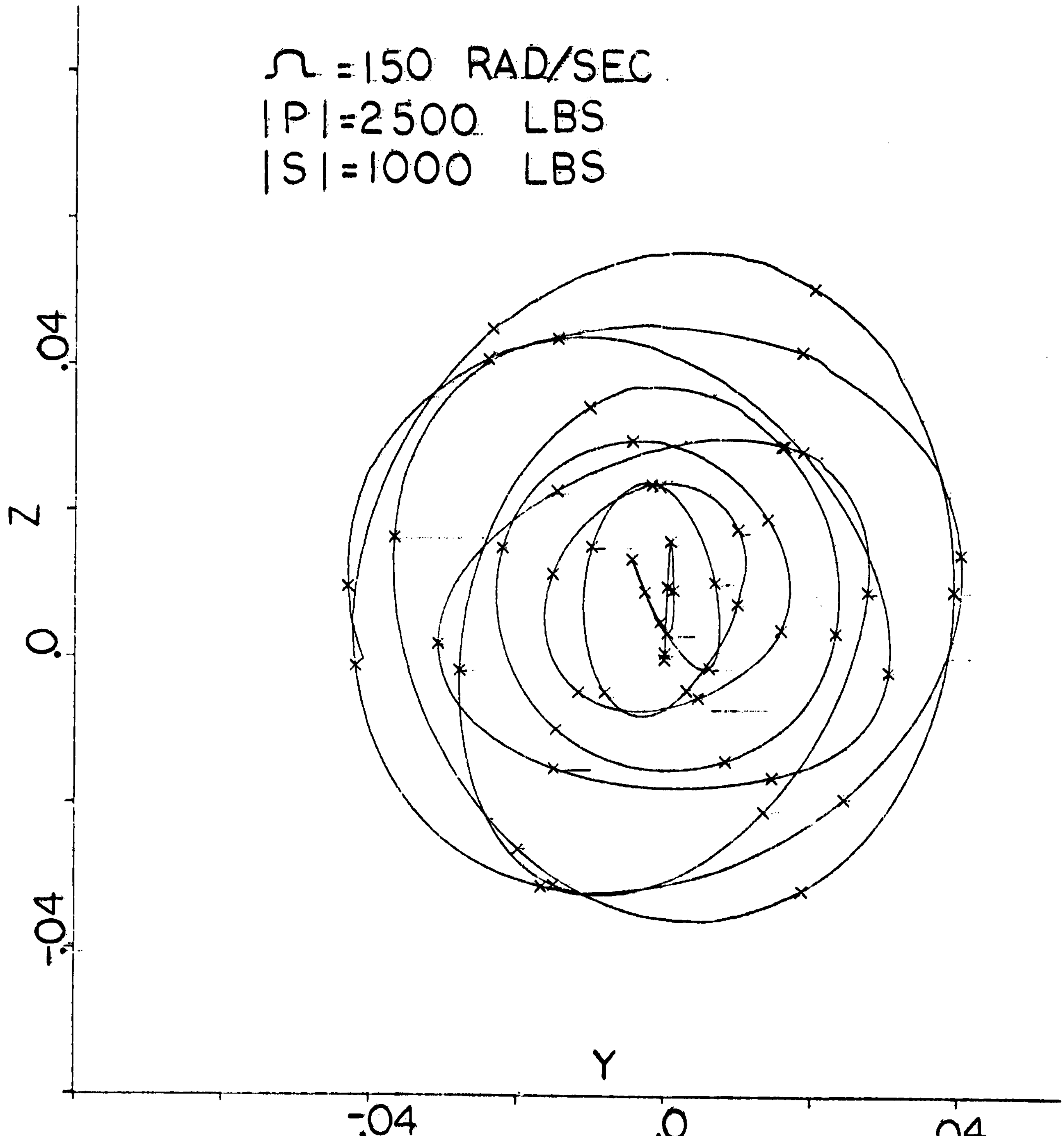


FIGURE 6.15 STATOR DISPLACEMENT
(SEE FIG. 6.1)

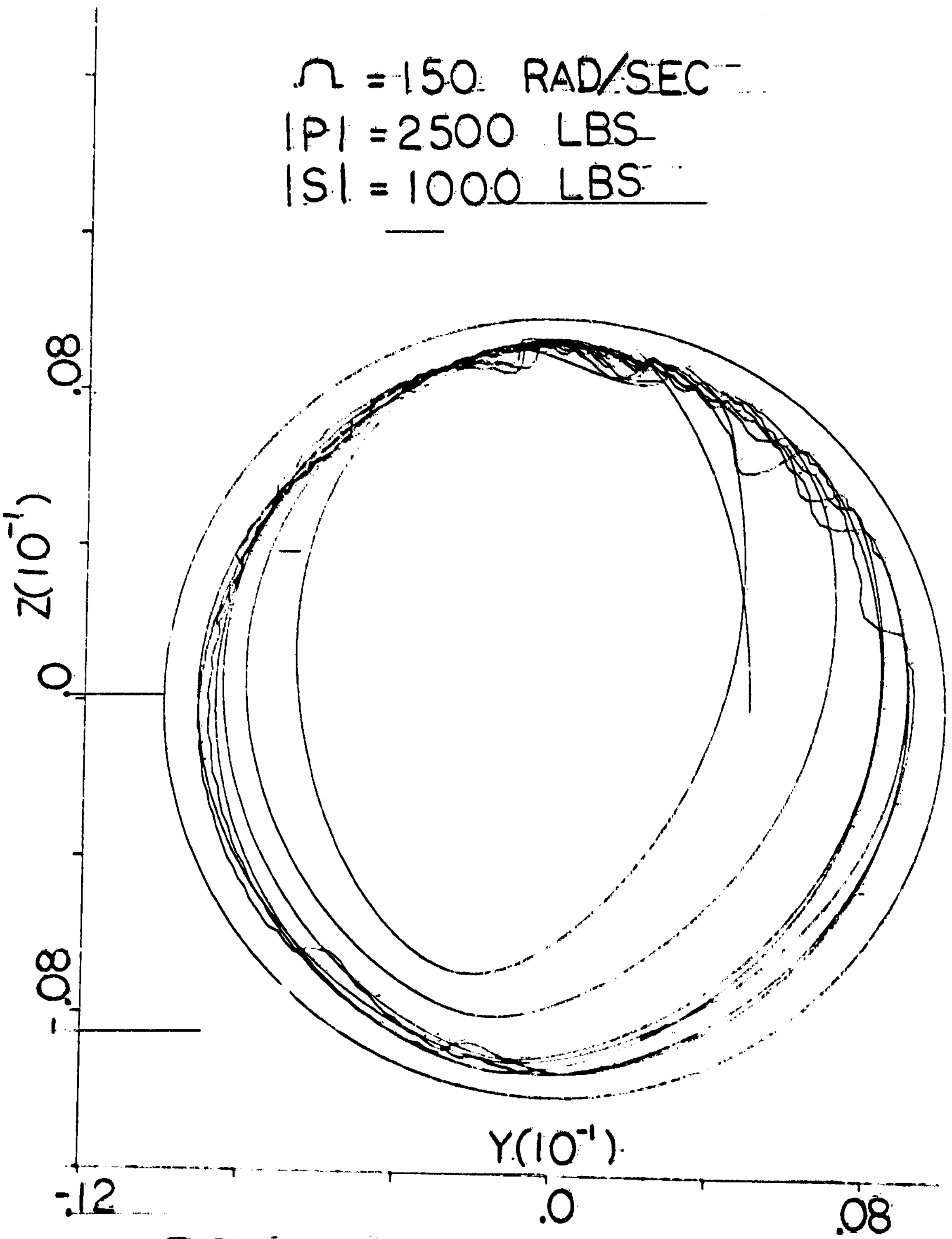
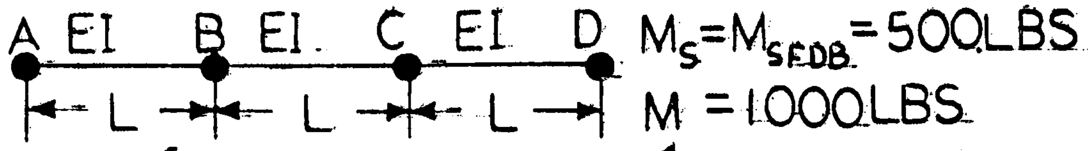


FIGURE 6.16 ROTOR ORBIT
(SEE FIG 6.1)



$E = 30.*10^6$ PSI, $I = 463.$ INCH^4, $L = 48$ INCH

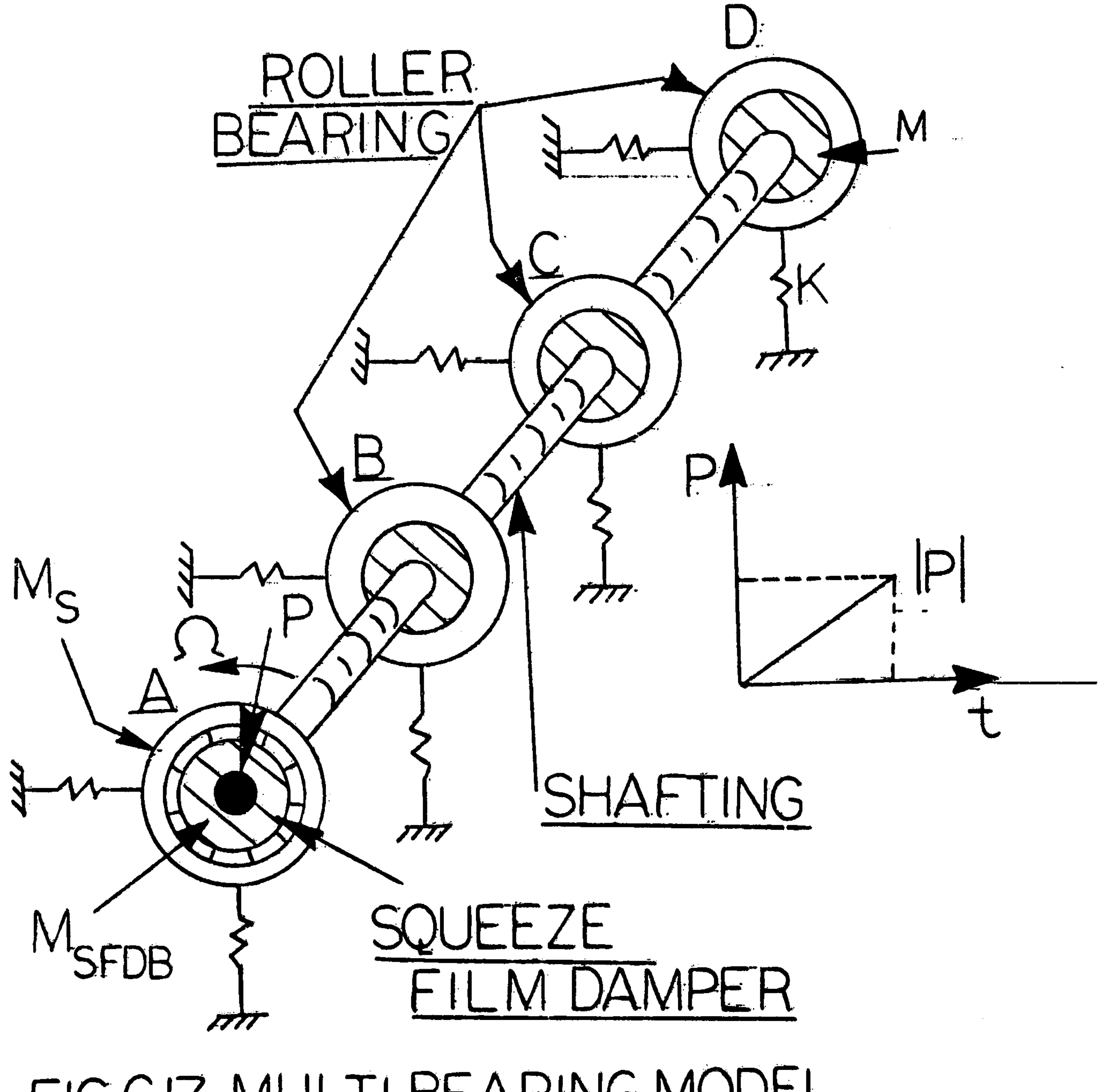


FIG. 6.17 MULTI BEARING MODEL

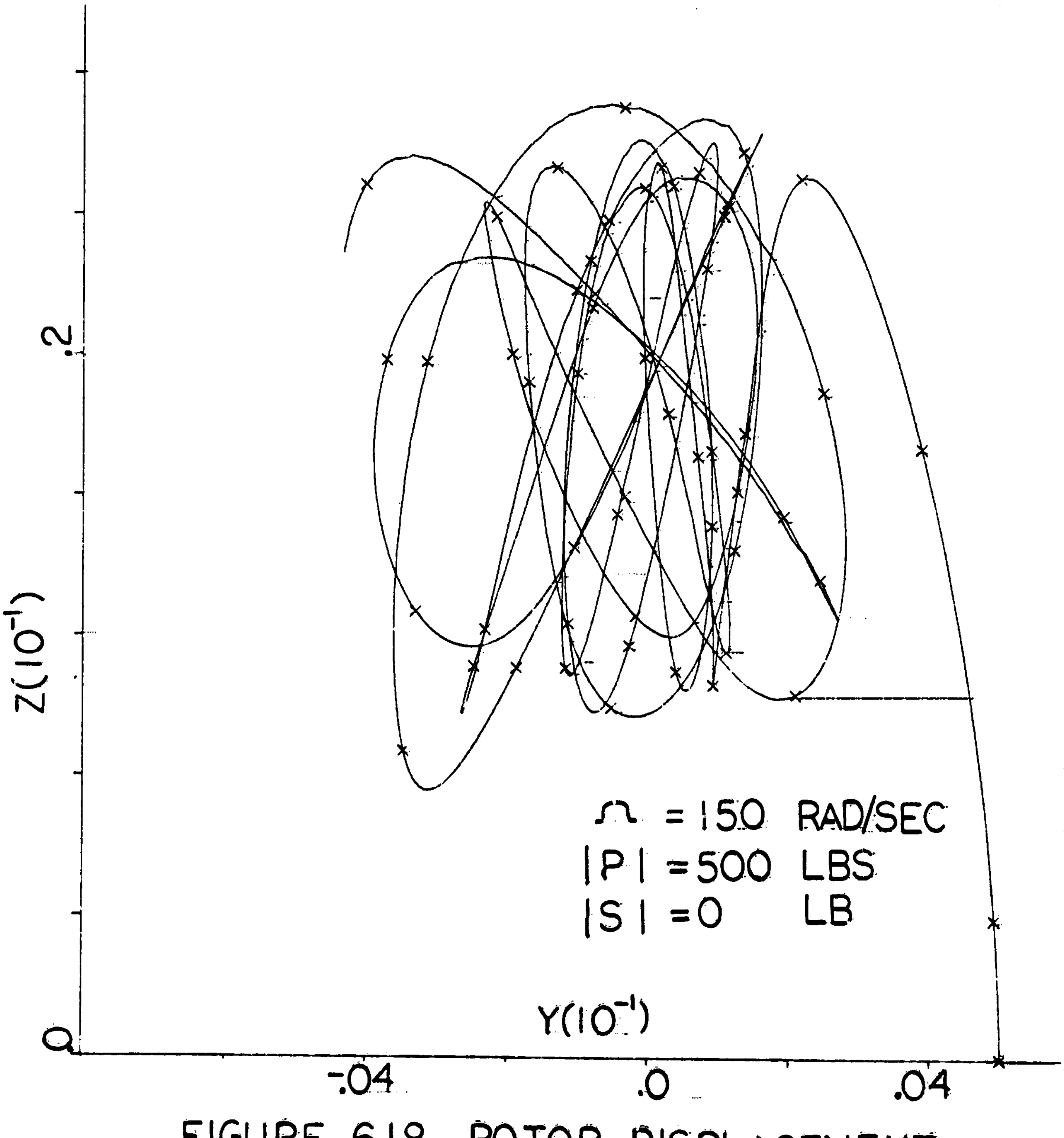


FIGURE 6.18 ROTOR DISPLACEMENT
(SEE FIG. 6.17)

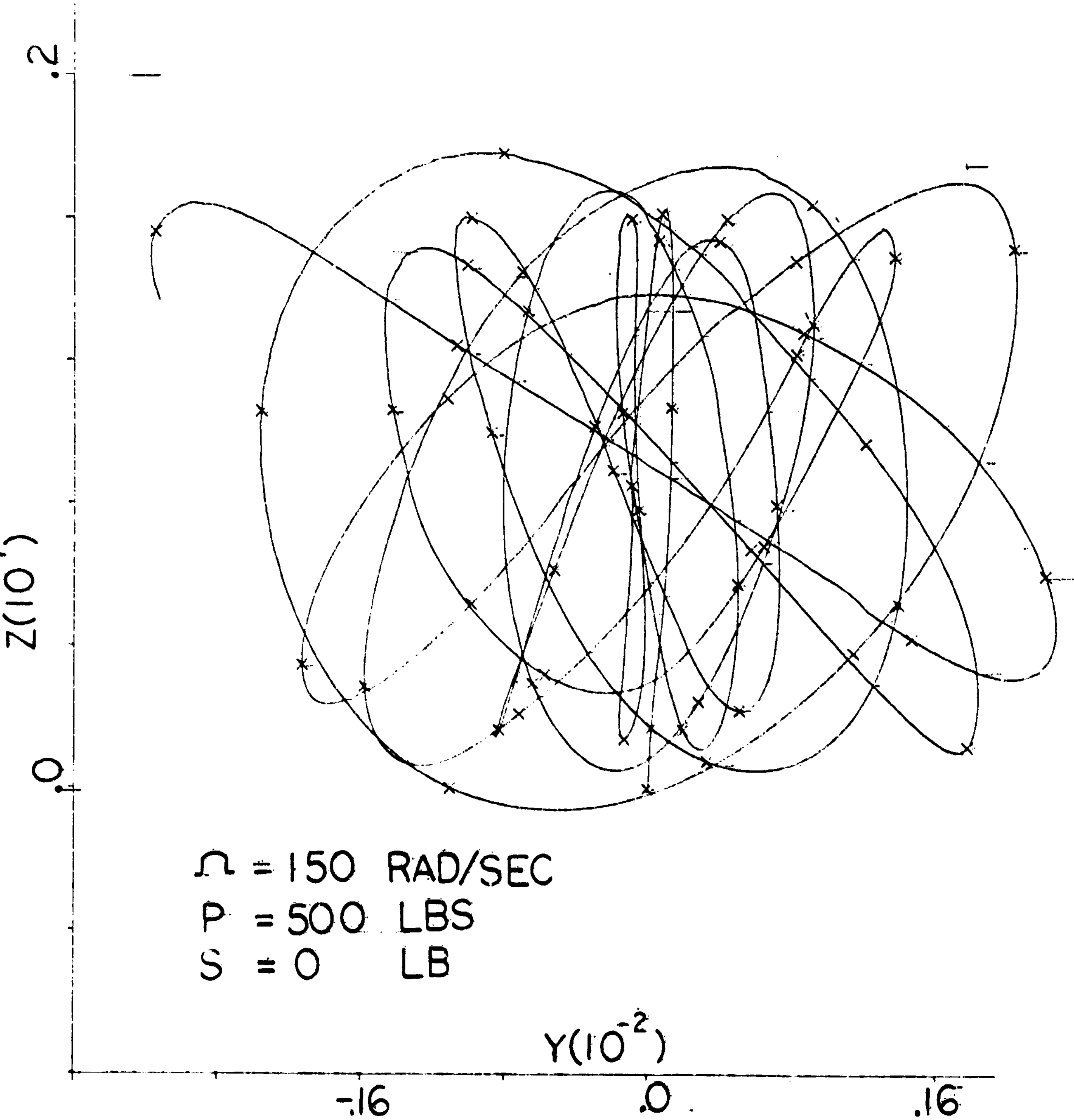


FIGURE 6.19 STATOR DISPLACEMENT
(SEE FIG. 6.17)

ORIGINAL PAGE IS
OF POOR QUALITY

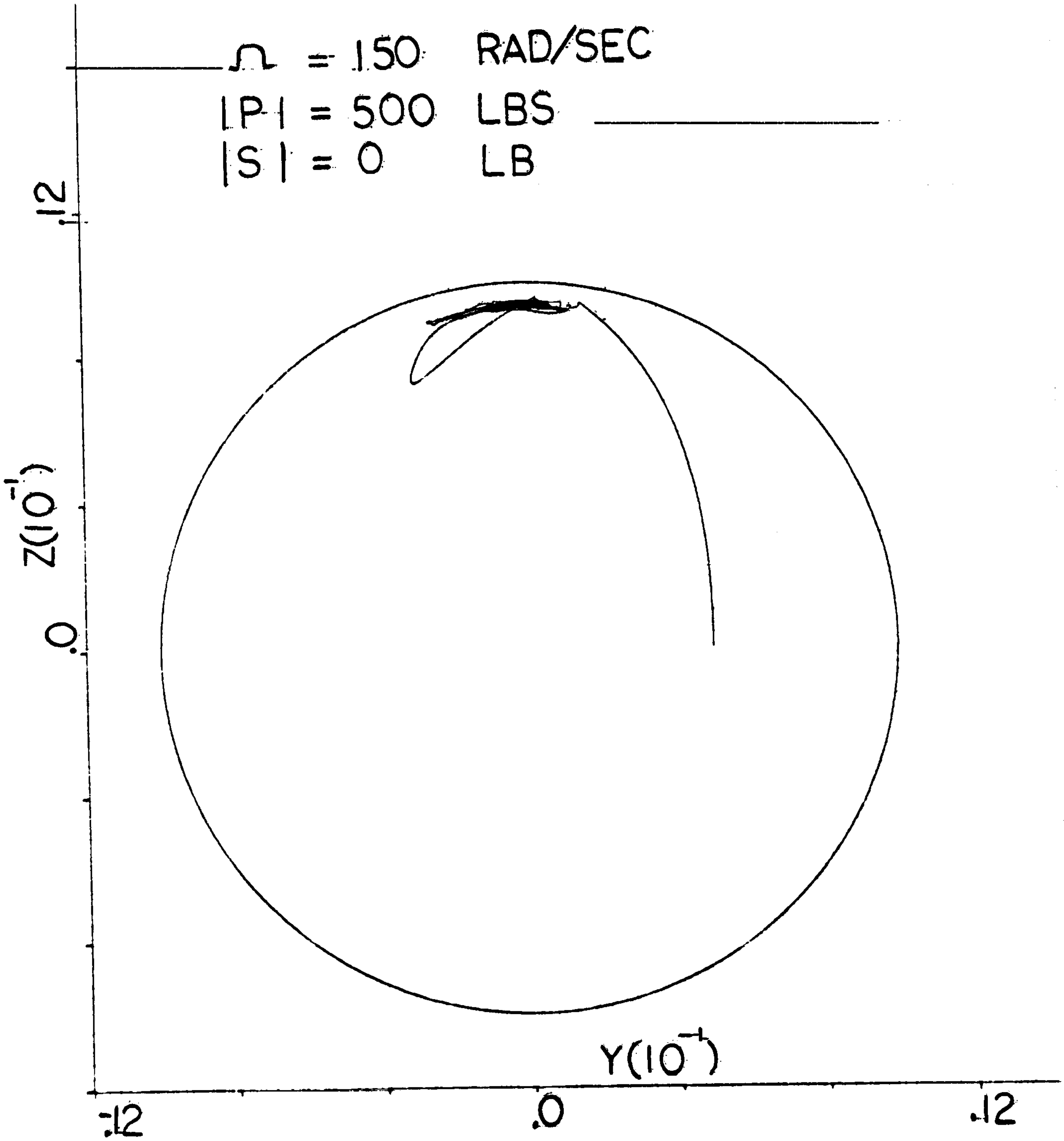


FIGURE 6.20 ROTOR ORBIT
(SEE FIG. 6.17)

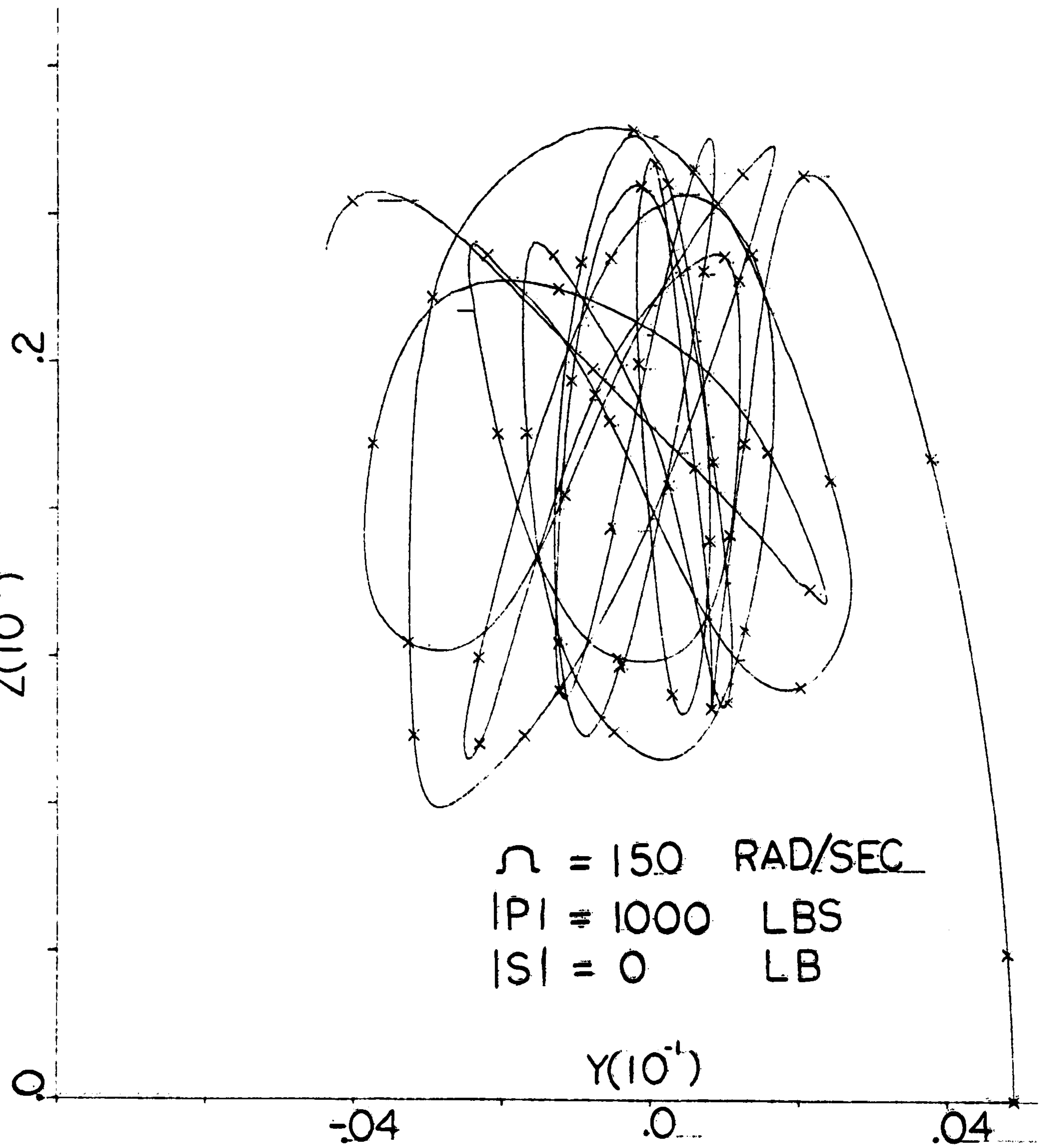


FIGURE 6.21 ROTOR DISPLACEMENT
(SEE FIG. 6.17)

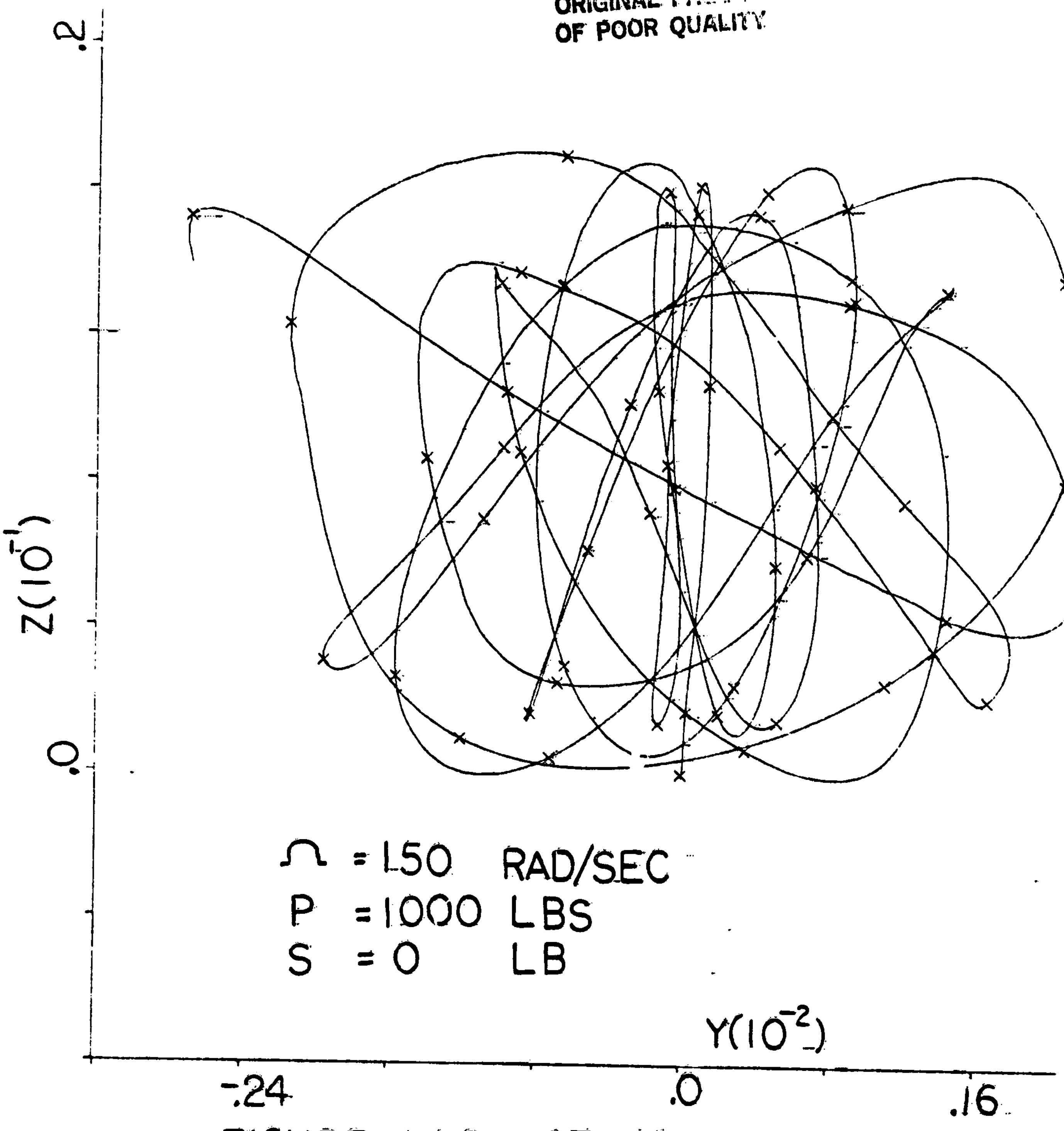


FIGURE 6.22 STATOR DISPLACEMENT
(SEE FIG. 6.17)

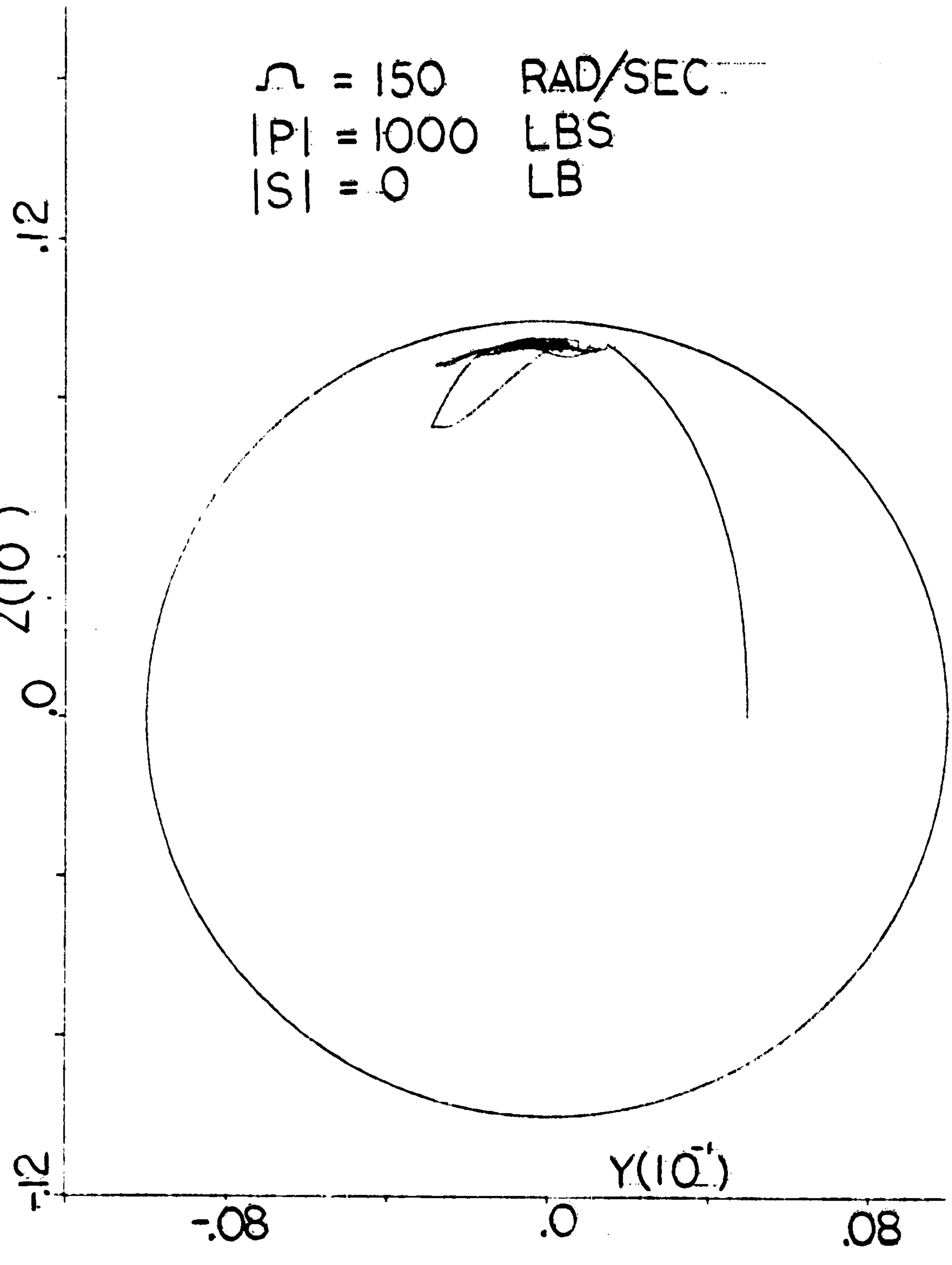


FIGURE 6.23 ROTOR ORBIT
(SEE FIG 6.17)

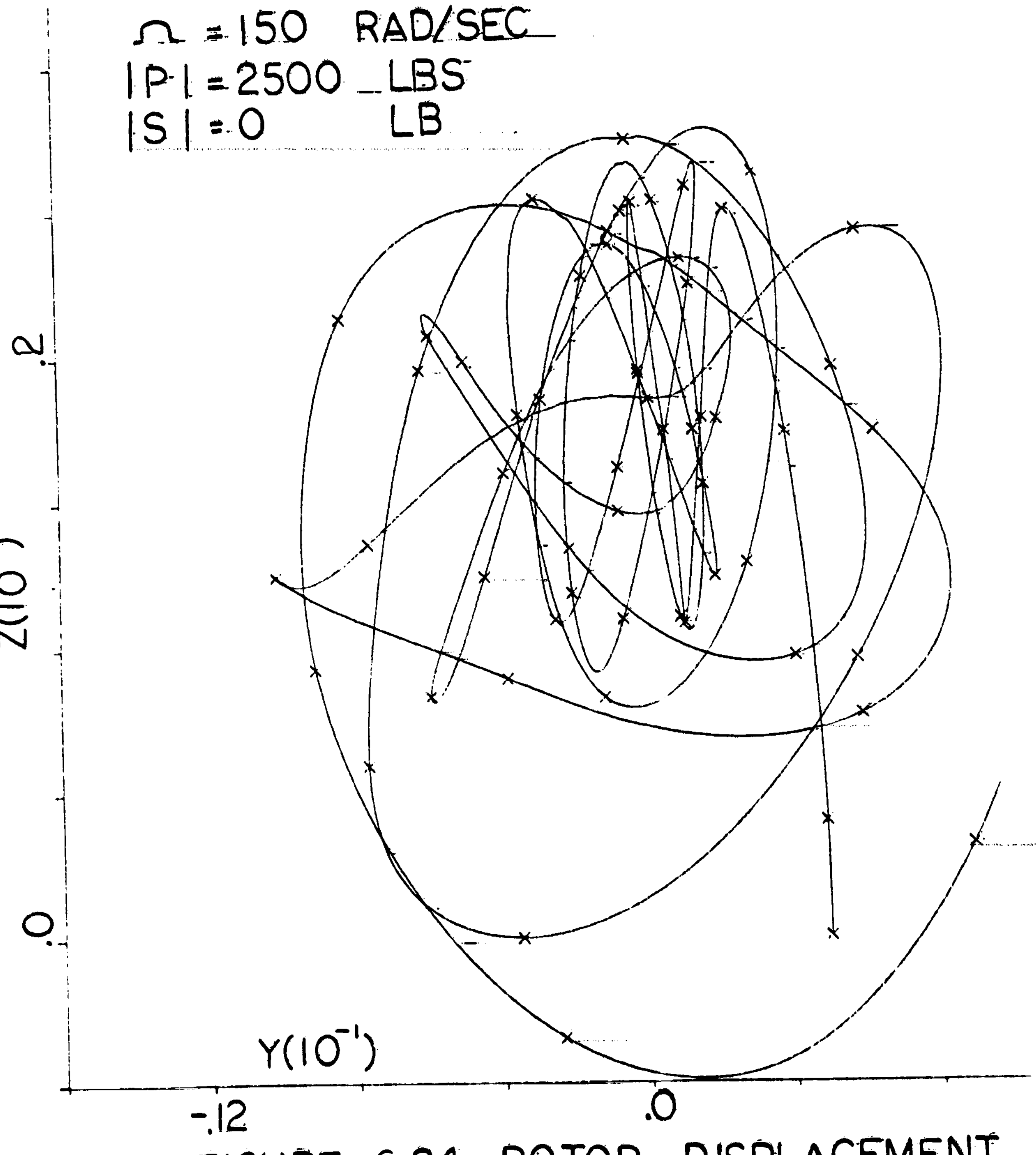


FIGURE 6.24 ROTOR DISPLACEMENT
(SEE FIG. 6.17)

ORIGINAL PAGE IS
OF POOR QUALITY

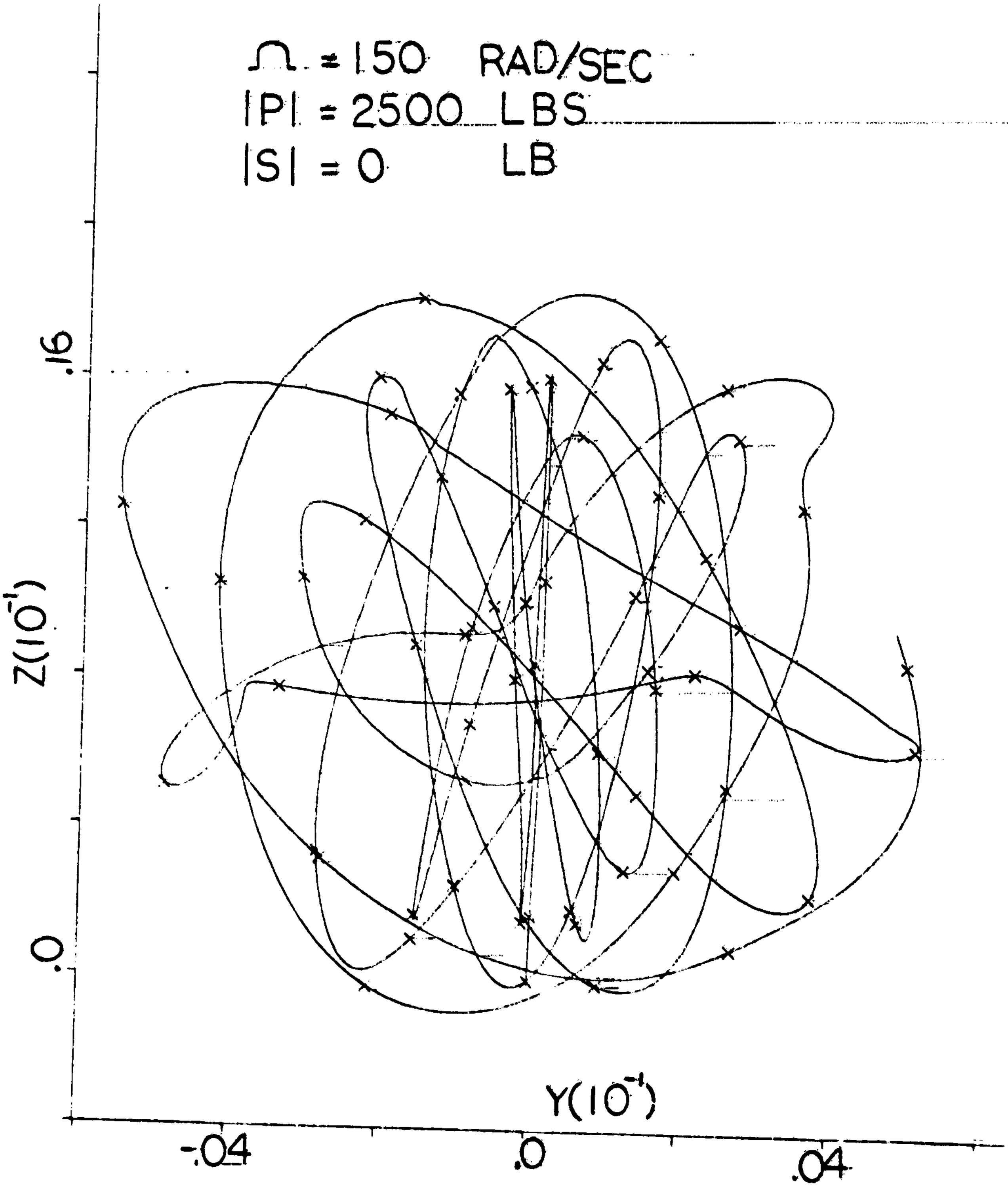


FIGURE 6.25 STATOR DISPLACEMENT
(SEE FIG. 6.17)

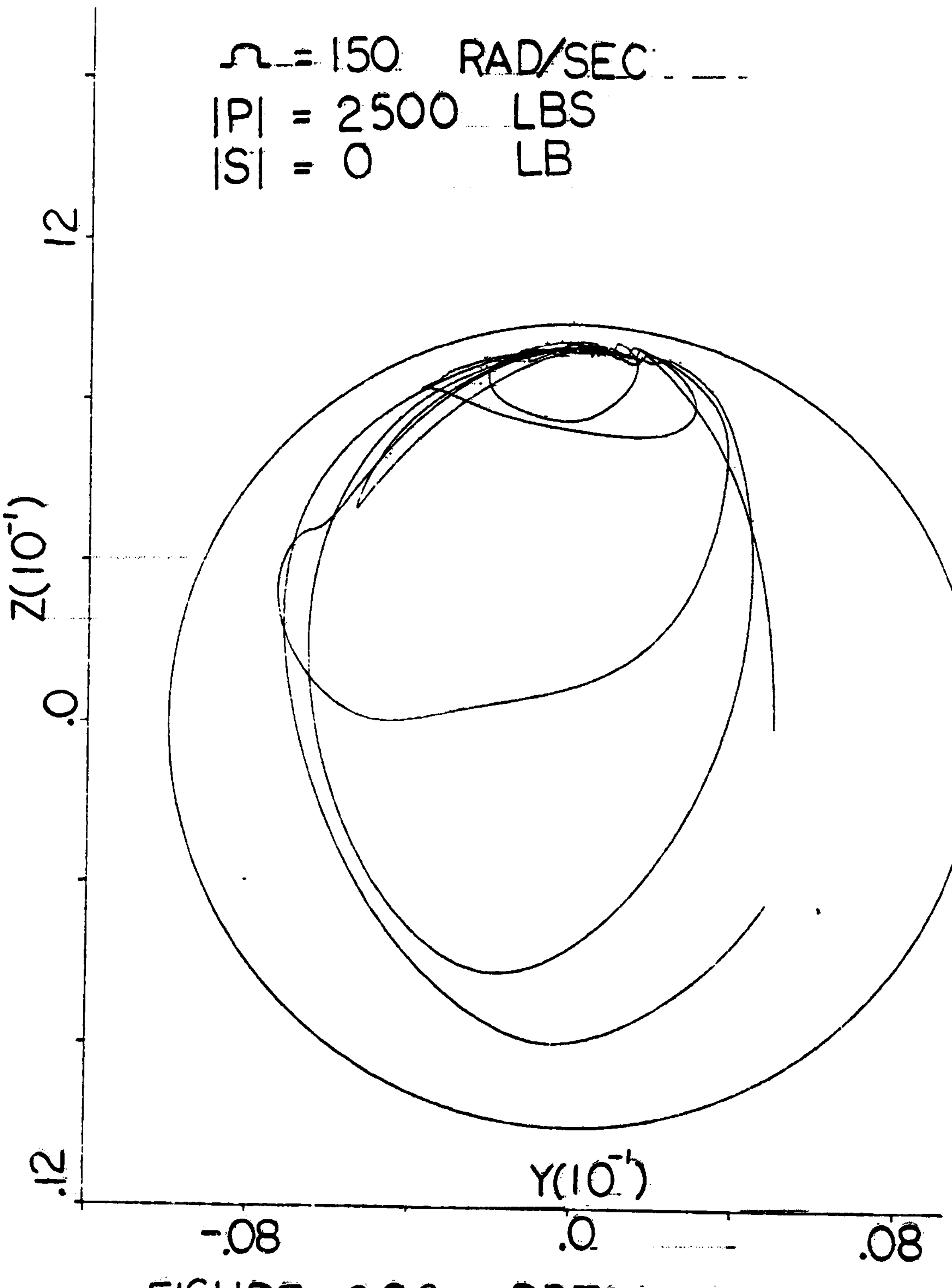


FIGURE 6.26 ROTOR ORBIT
(SEE FIG. 6.17)

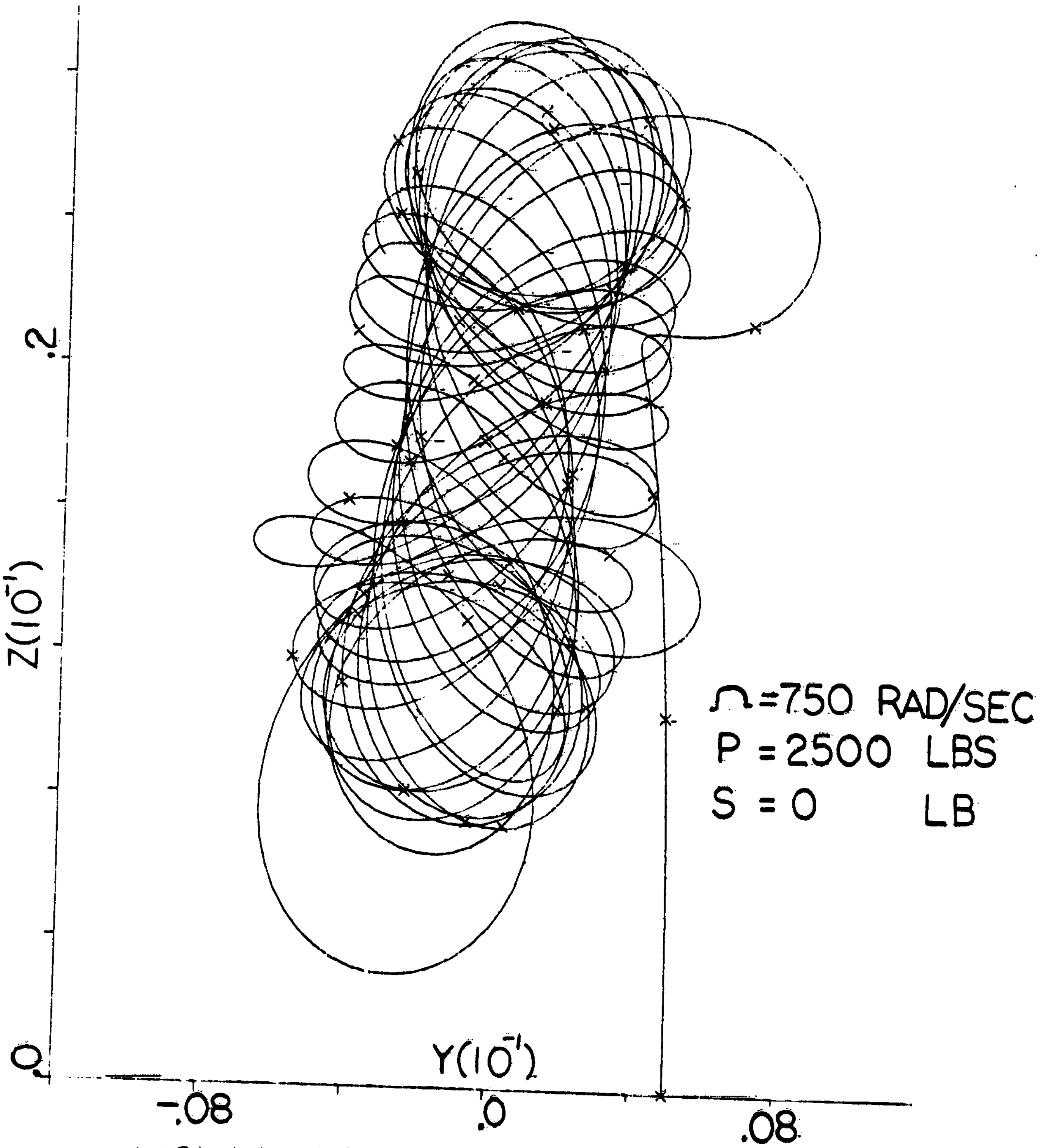


FIGURE 6.27 ROTOR DISPLACEMENT
(SEE FIG 6.17)

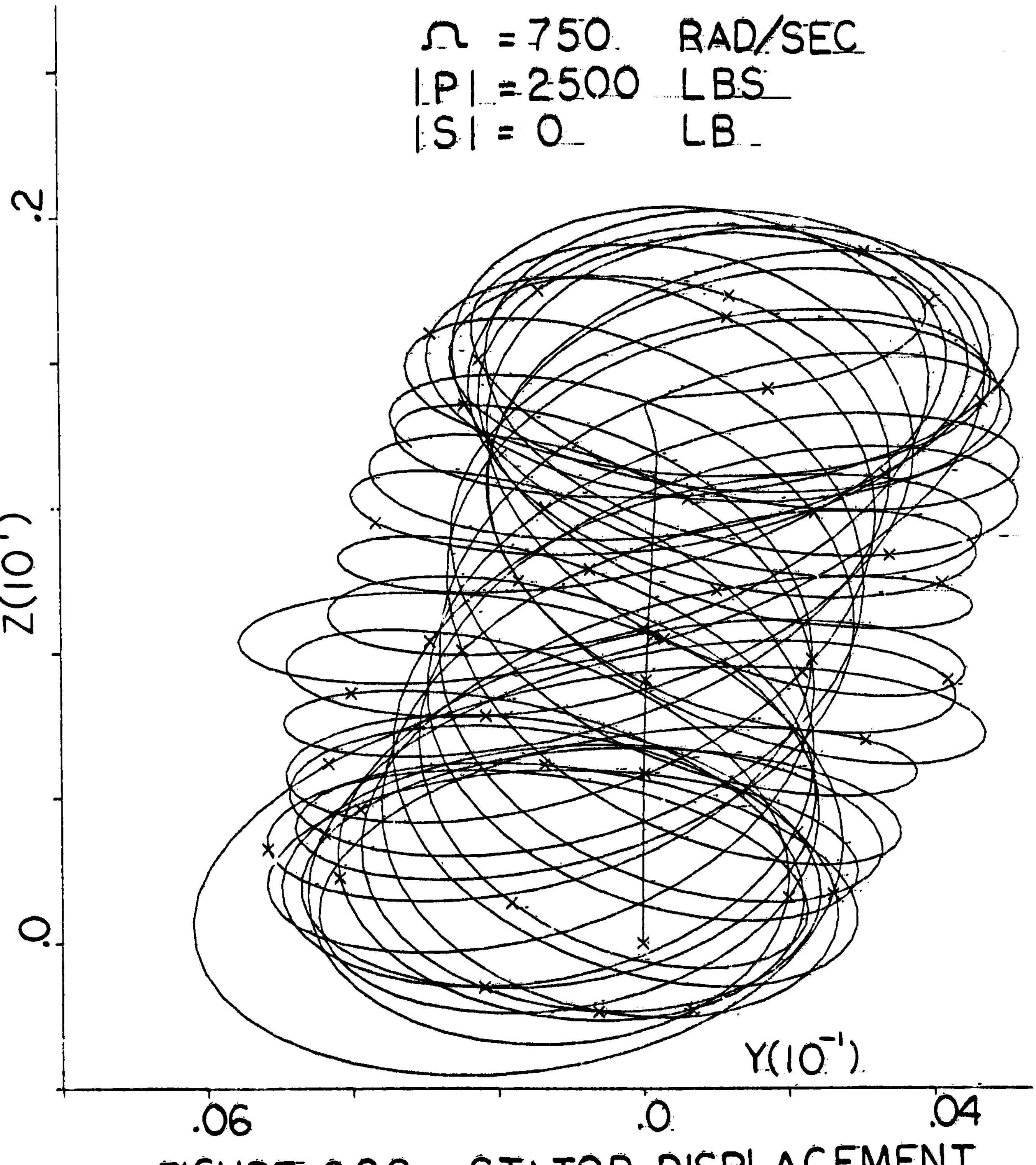


FIGURE 6.28 STATOR DISPLACEMENT
(SEE FIG. 6.17)

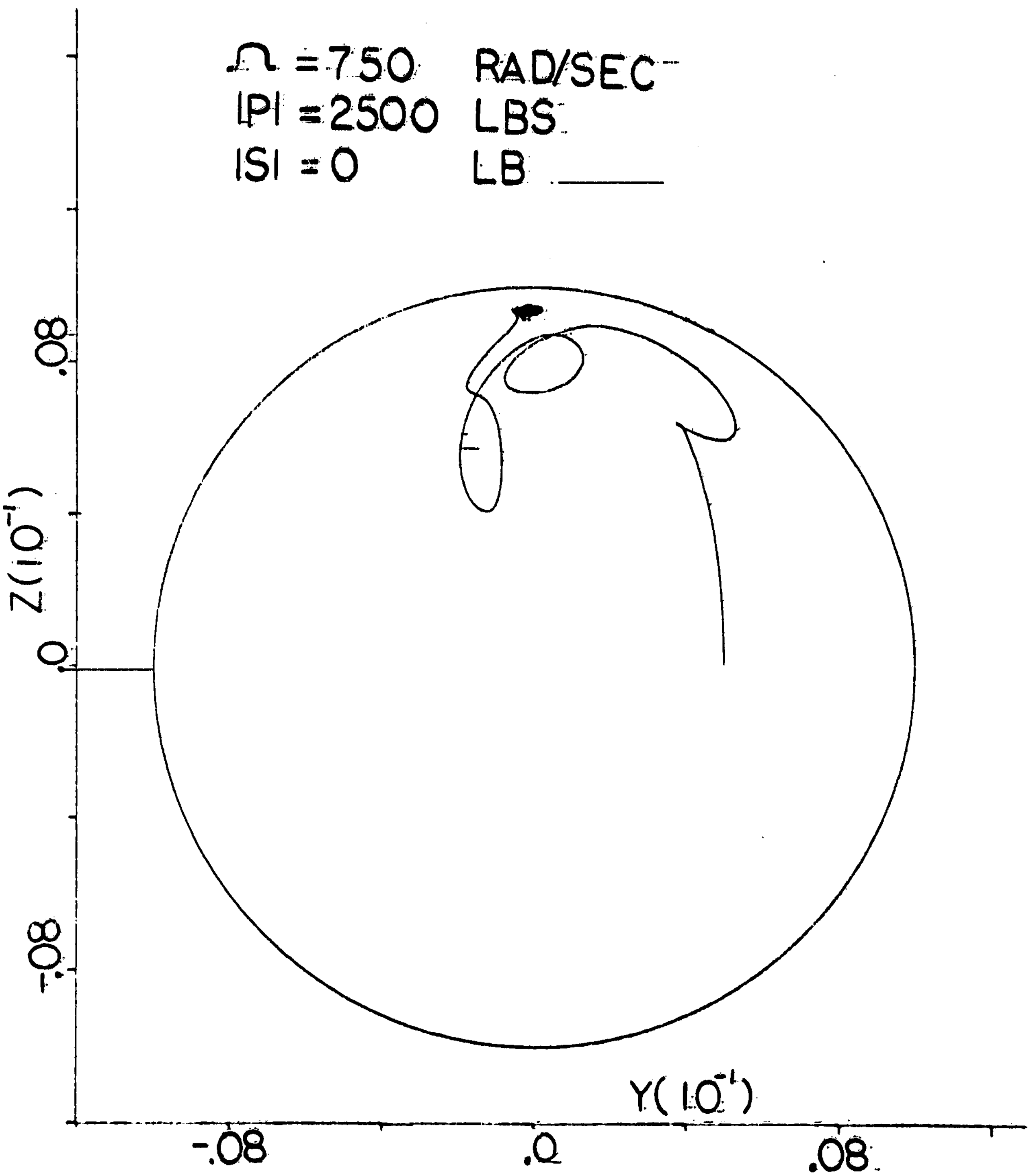


FIGURE 6.29 ROTOR ORBIT
(SEE FIG. 6.17)

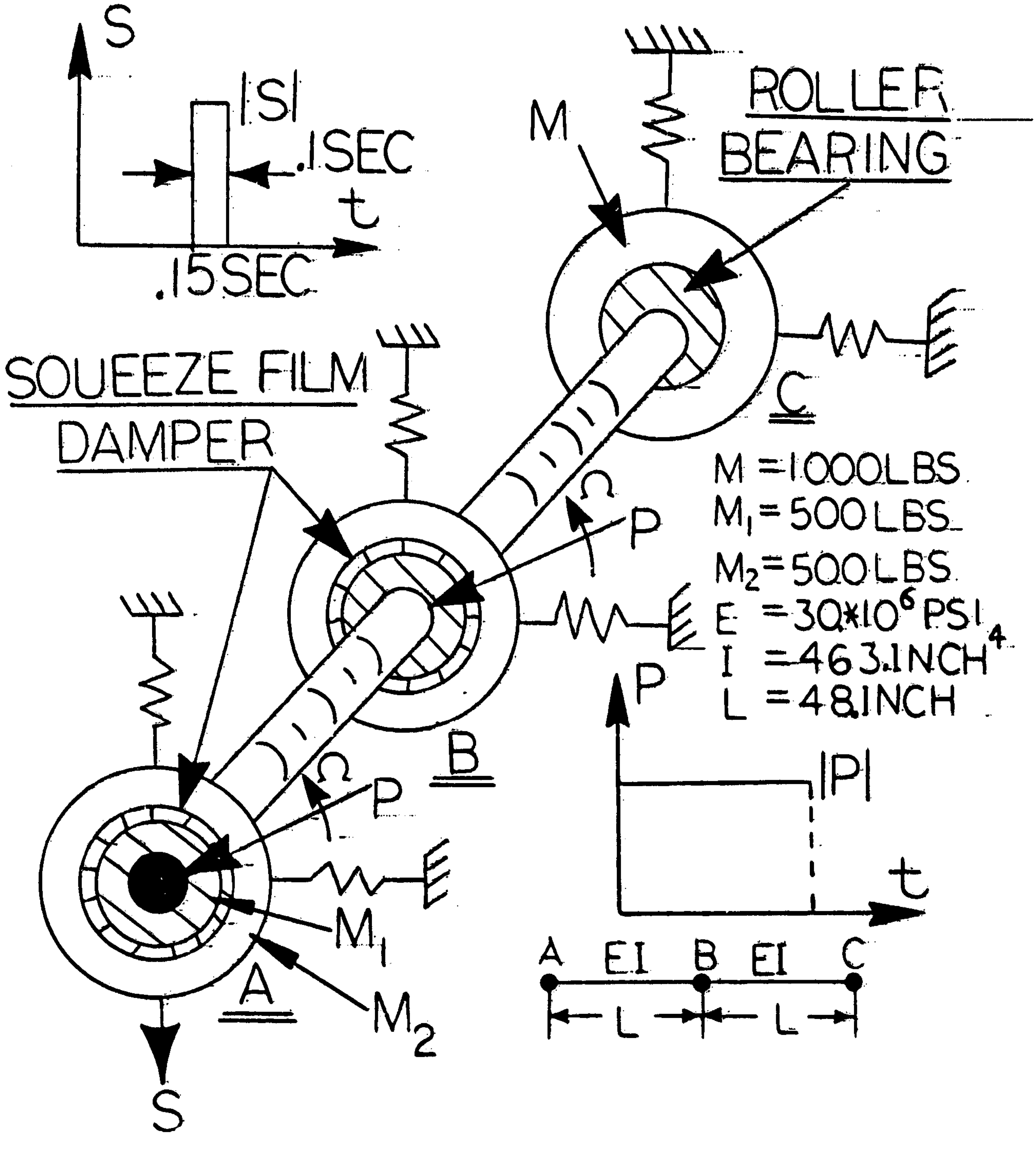


FIG. 6·30 MULTI BEARING MODEL

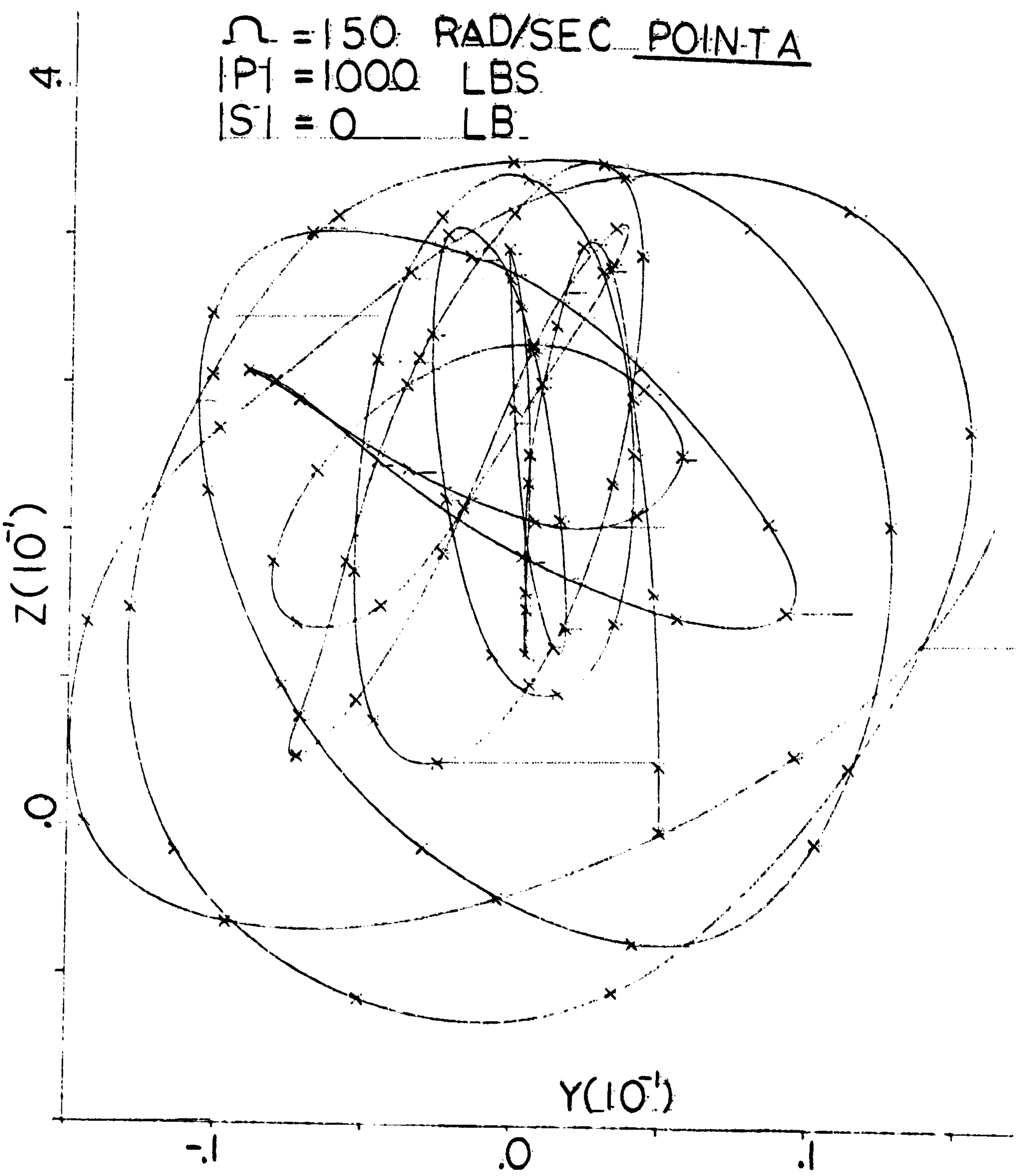


FIGURE 6.31 ROTOR DISPLACEMENT
(SEE FIG. 6.30)

ORIGINAL PAGE IS
OF POOR QUALITY

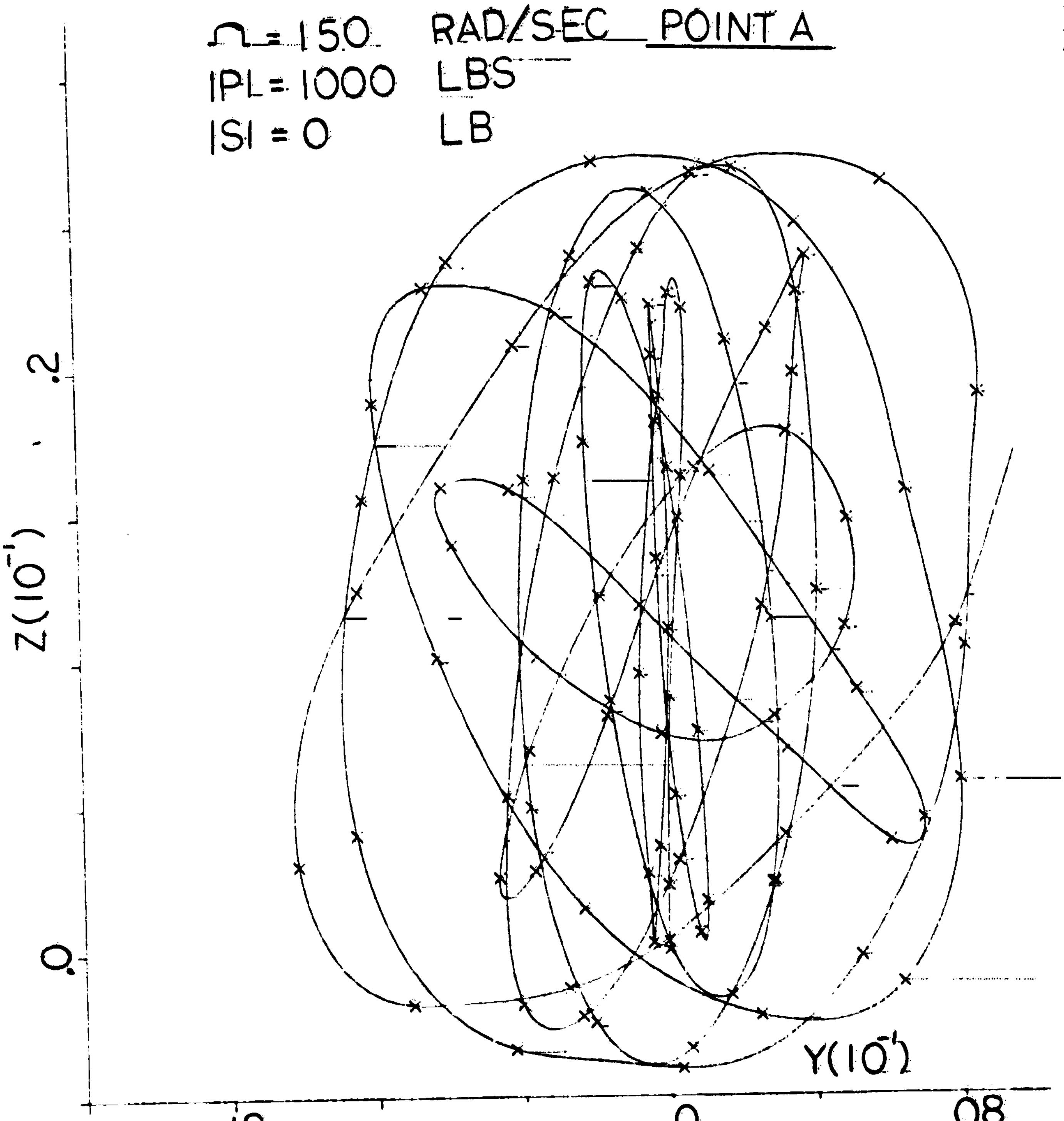


FIGURE 6.32 STATOR DISPLACEMENT
(SEE FIG. 30)

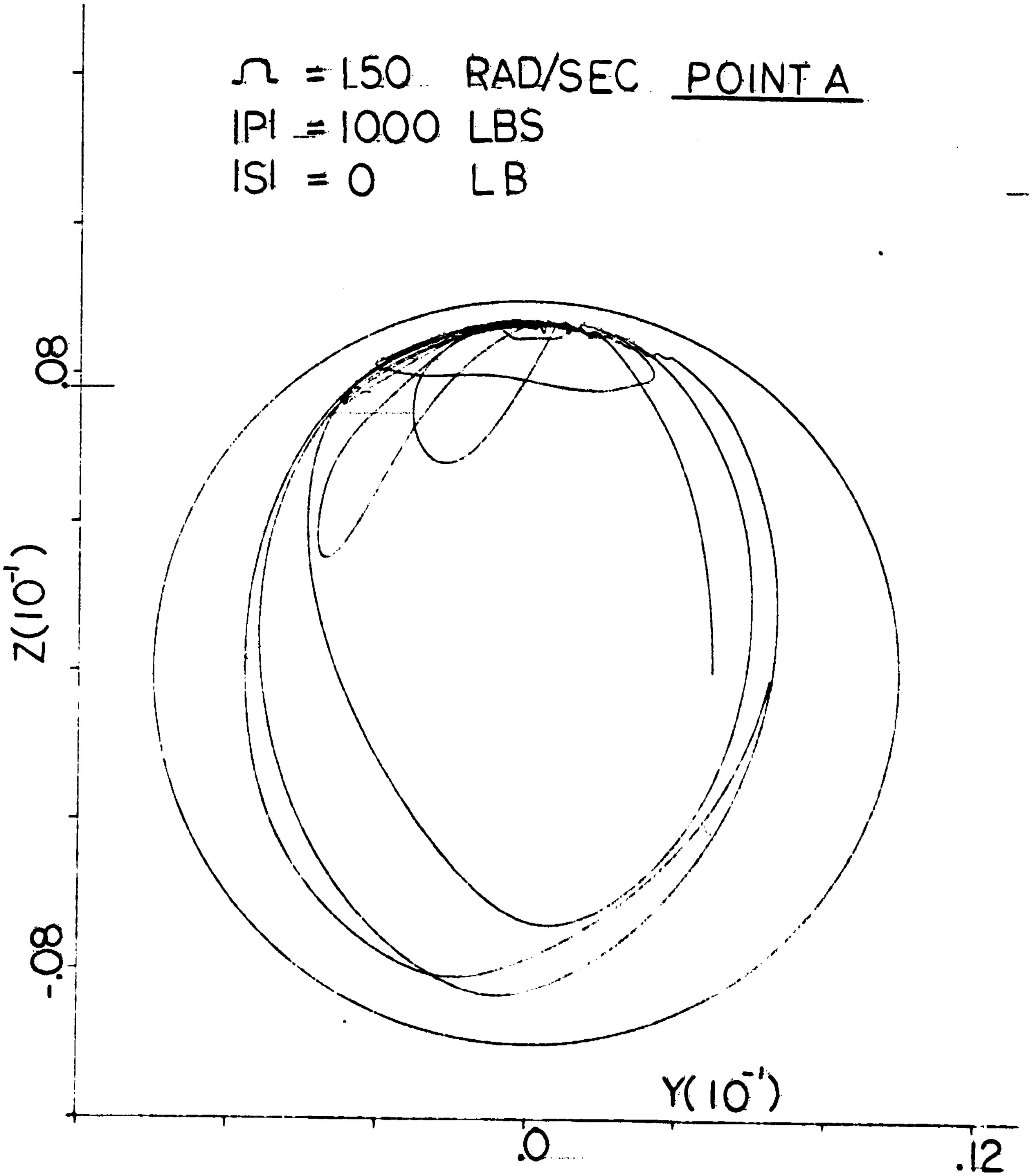


FIGURE 6.33 ROTOR ORBIT
(SEE FIG. 30)

ORIGINAL PAGE IS
OF POOR QUALITY

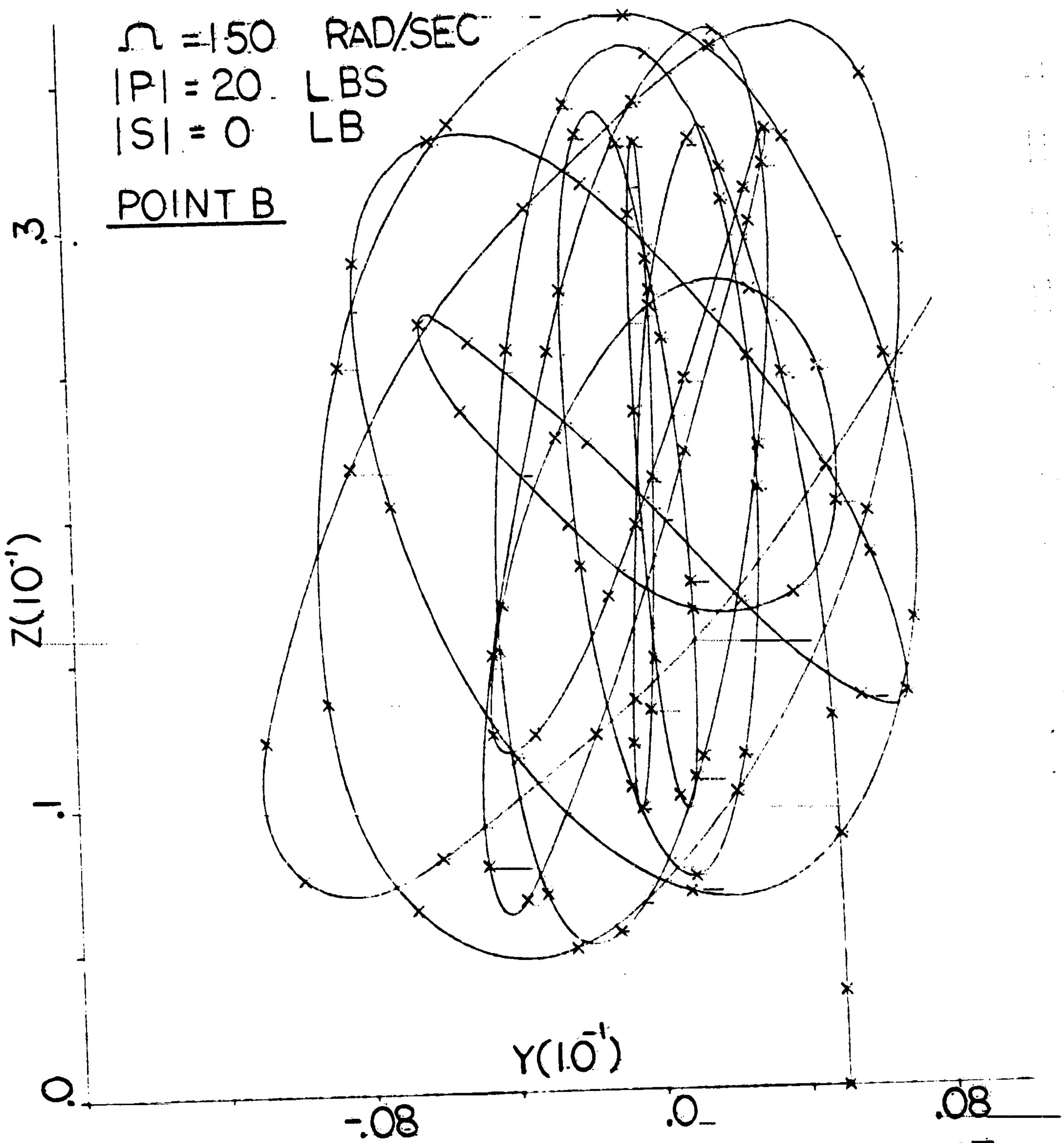


FIGURE 6.34 ROTOR DISPLACEMENT
(SEE FIG. 6.30)

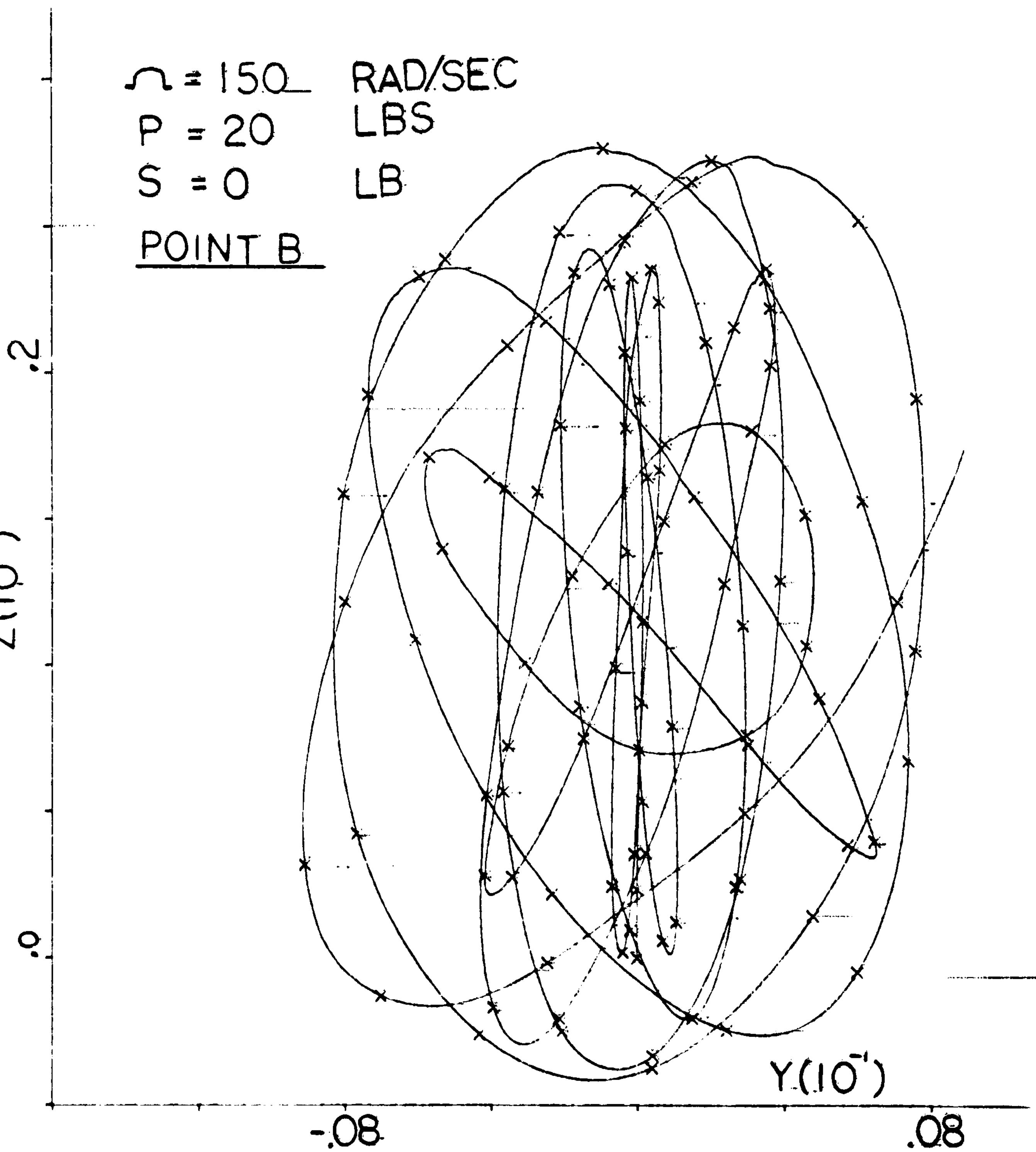


FIGURE 6.35 STATOR DISPLACEMENT
(SEE FIG. 6.30)

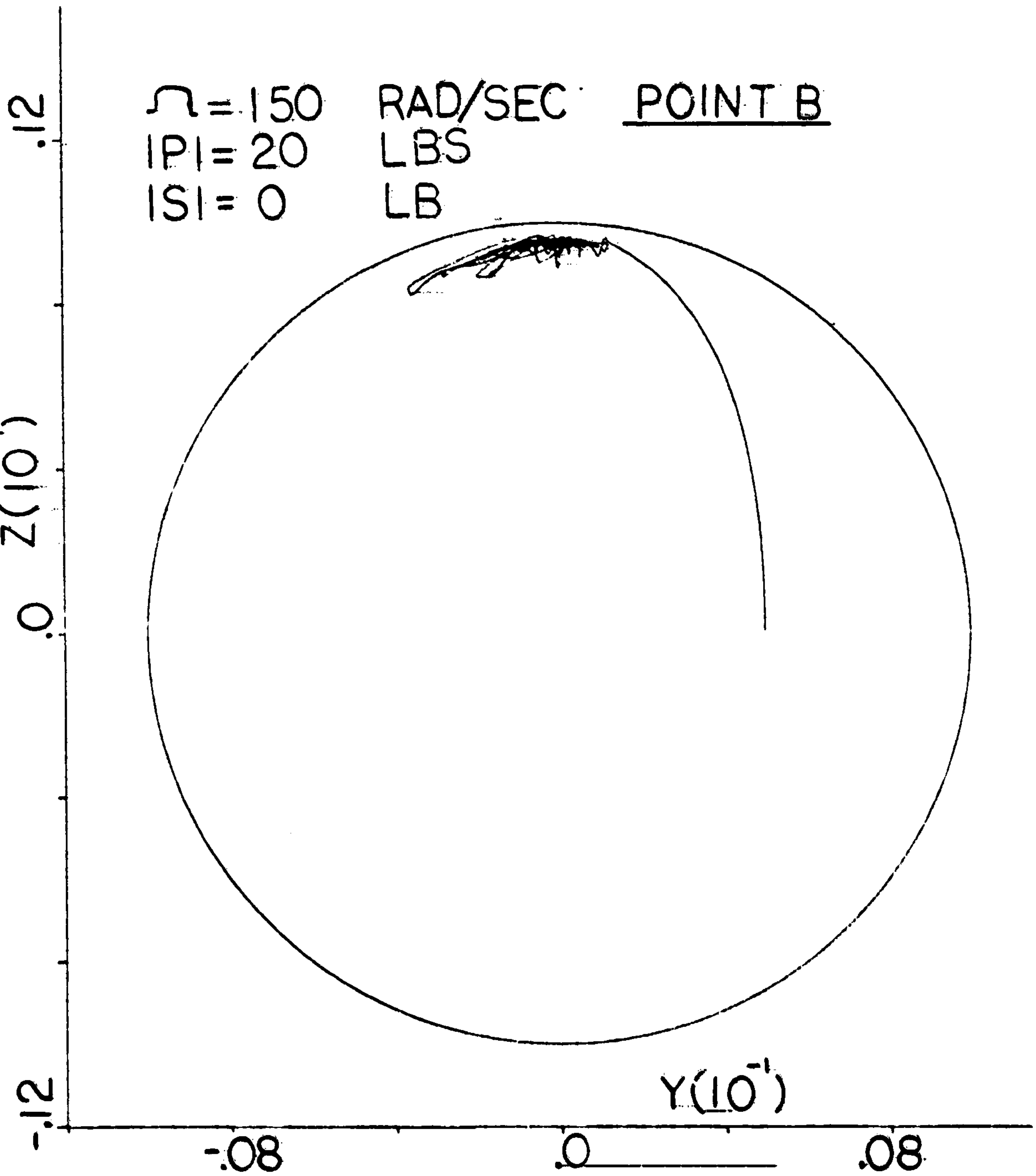


FIGURE 6.36 ROTOR ORBIT
(SEE FIG. 6.30)

APPENDIX 2: USER I/O INFORMATION

This appendix outlines a supplement that must be employed together with the main "ADINA" manual to help model and run general engine dynamic simulations. The manual presents the necessary changes and additions to "ADINA" to implement the bearing element into the code.

To simplify its use, the original flow of the ADINA input remains basically the same. This is achieved by adding a new bearing element to the I/O stream. This was made possible by introducing several new features to the I/O namely:

(1) A master control card;

(2) A bearing element group;

(3) A rotating inbalance loads generator; and,

(4) Plotting capabilities.

In what follows, items i) - iv) will be discussed in detail. As noted earlier, such information is to be used in conjunction with the standard ADINA I/O manual. In keeping with the style of that manual, the following discussion is written in the same form.

A2.1 Master Control Card

Card 3 in ADINA user's manual should be replaced by the following card:

II MASTER CONTROL CARDS (continued)

Card 3: Frequency Solution Request Card (2I5)

note	columns	variable	entry
(1)	1 - 5	IEIG	Flag indicating frequency solution mode; EQ.0; no frequency solution EQ.1; find lowest frequencies and associated mode shapes

note	columns	variable	entry
(2)	6 - 10	IBEAR	Flag indicating bearing dynamic simulation; EQ.0; no bearing simulation required EQ.1; bearing simulation required

NOTES/

(1) See ADINA user's manual.

(2) The control parameter IBEAR determines if the program is to simulate squeeze film damper employing the bearing element. If IBEAR equal to 1, then bearing element group must be input.

A2.2 Bearing Element Group

Bearing elements are members allowed arbitrary orientation in the global coordinate system used in ADINA. The bearing element is 2-node element with only global translational degrees of freedom at each node. The bearing element simulates the squeeze film damper. One of the two nodes of the element represents the damper while the second node represents the stator.

The following input is required for each element group consisting of bearing elements:

Section A Element Group Control Card

This card defines the number of elements in this group, etc.

Section B Element Data Cards

1. Geometric and fluidic properties of squeeze film damper. The nominal damper annulus diameter, length and radial clearance are defined in this section for each bearing element. The positions of the damper ports, the pressure

at each port, the lubricant viscosity, and the film rupture pressure are also defined. Other variables related to the calculations of the damper tangent stiffness and damping matrices are defined in this section.

2. Element Nodes: The nodal points of bearing elements are input in this part.

A specific element group defined by the above input cards is followed by the input cards of another element group.

A2.2A <u>Element Group Control Card</u> (6I4)

note	columns	variable	entry
	1 - 4	NPAR(1)	Enter number "1"
(1)	5 - 8	NPAR(2)	Number of bearing elements; GE.1
(2)	9 - 12	NPAR(3)	Enter the number "1"
(3)	13 - 16	NPAR(4)	Element birth and death option; EQ.0 elements are active throughout the solution (see note 3)
(4)	17 - 20	NPAR(5)	Element type code; Enter the number "3"
(5)	25 - 28	NPAR(7)	Maximum number of nodes used in any one element; Enter the number "2"

NOTES/

(1) BEARING element numbers in each group begin with one (1) and end with the total number of elements in the group.

(2) The parameter NPAR(3) is used to help the program to go through the proper execution path.

(3) The variable NPAR(4) identifies whether the elements of the element group are active throughout the solution. NPAR(4) can take different modes. For the present time, it takes the value 0. If NPAR(4) .EQ. 0, the elements are active throughout the solution.

(4) The variable NPAR(5) defines the type of the element used in this element group. Currently, it only takes the value "3".

(5) NPAR(7) limits the number of nodes that can be used to describe any bearing element in this element group.

A2.2B Element Data Cards

A2.2B.1 Geometric and Fluidic Properties of Bearing Element

The necessary pre-selected geometric parameters of the damper is read in this section. This includes the annulus diameter of the damper, the bearing length, and the bearing clearance. The oil properties (e.g., viscosity and pressure rupture) are also read. The element selection, stiffness matrix formation, and the damping matrix formation options are input in this section also.

Card 1: (I10,6f10.0)

note	columns	variable	entry
(1)	1 - 10	M	Bearing element number; GE.1 and LE.NPAR(2)
	11 - 20	BD	Nominal bearing annulus diameter
	21 - 30	BL	Nominal bearing length
	31 - 40	BC	Bearing annulus radial clearance
	41 - 50	VISC	Bearing lubricant viscosity
	51 - 60	PVAP	Film rupture absolute pressure (PSIA)
(2)	61 - 70	TH1	Position angle of lubricant port-1 (degrees)

Card 2: (3f10.0,6I5)

note	columns	variable	entry
(2)	1 - 10	TH2	Position angle of lubricant port-2 (degrees)
	11 - 20	PB1	Specific boundary pressure at port-1 (PSIA)
	21 - 30	PB2	Specific boundary pressure at port-2 (PSIA)
	31 - 35	NGRID	Number of finite-difference grid points per damper (odd)
	36 - 40	NSOLN	Solution type; EQ.1 long-bearing solution EQ.2 short-bearing solution EQ.3 Fourier-Series 2-D convergent solution
	41 - 45	NPORT	Number of lubricant ports; GE.0 and LE.2

ORIGINAL PAGE IS
OF POOR QUALITY

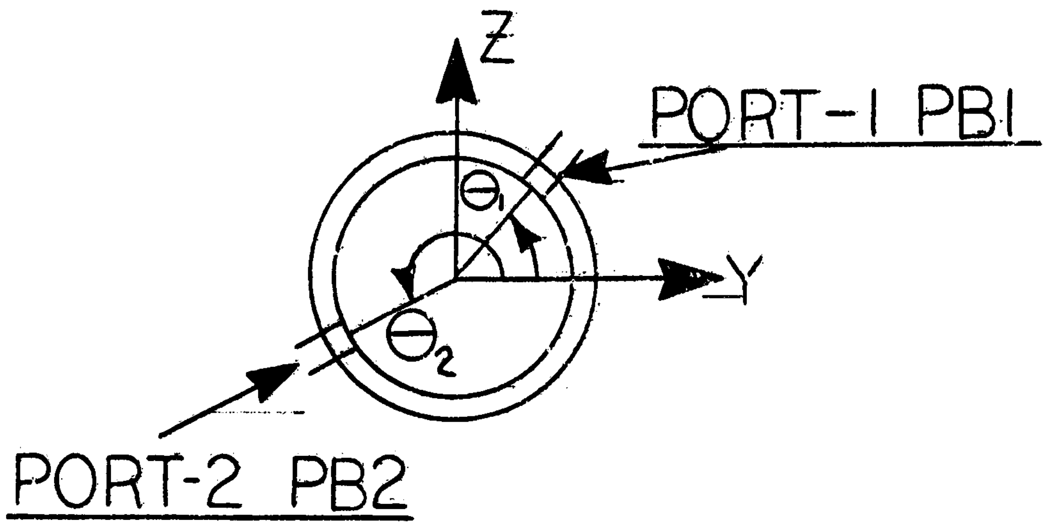
Card 2: (3f10.0,6I5) continued...

note	columns	variable	entry
	46 - 50	NFILM	Number of identical annuli for the bearing
	51 - 55	KOFK	Bearing element stiffness matrix formation option; EQ.0 no stiffness reformation is performed EQ.1 stiffness reformation is performed.
	56 - 60	KOEC	Bearing element damping matrix formation option; EQ.0 no damping matrix formation is performed. EQ.1 damping matrix is performed.

NOTES/

(1) Elements must be input in increasing element number order. If cards for elements (M+1, M+2,...,M+J) are omitted, these "J" missing elements can be generated in the same way as ADINA does.

(2) The position angles of lubricant ports must be input in degrees and are measured as shown below:



A2.2B.2 Element Nodes

Following the previous bearing information, each elements' nodes and their degrees of freedom are input. The degrees of freedom of the nodes which represent the bearing element may be different from those defined in master control card number "1". Accordingly, the degrees of freedom of the bearing element nodes are redefined as applied to this element individually.

Card 3: (15I4)

note	columns	variable	entry
	1 - 4	M	Bearing element number; GE.1 and LE. NPAR(2)
	5 - 8	IELD	Number of nodes to describe this element
	9 - 12	KG	Node generation increment used to compute the node number for missing elements; EQ.0 default set to "1"
(1)	13 - 16	IDOFB(1)	Rotor node X-translation code; EQ.0: admissible EQ.1: deleted
(1)	17 - 20	IDOFB(2)	Rotor node Y-translation code
(1)	21 - 24	IDOFB(3)	Rotor node Z-translation code
(1)	25 - 28	IDOFB(4)	Rotor node X-rotation code
(1)	29 - 32	IDOFB(5)	Rotor node Y-rotation code
(1)	33 - 36	IDOFB(6)	Rotor node Z-rotation code

Card 3: (15I4) continued...

note	columns	variable	entry
(1)	37 - 40	IDOFB(1)	Stator node X-translation code; EQ.0: admissible EQ.1: deleted
(1)	41 - 44	IDOFB(2)	Stator node Y-translation code
(1)	45 - 48	IDOFB(3)	Stator node Z-translation code
(1)	49 - 52	IDOFB(4)	Stator node X-rotation code
(1)	53 - 56	IDOFB(5)	Stator node Y-rotation code
(1)	57 - 60	IDOFB(6)	Stator node Z-rotation code
(1)	61 - 64	IPLOTE	Control parameter for saving bearing responses for future plotting; EQ.0: bearing responses are not saved EQ.1: responses are saved on tape to be used by post-processor 2-D and 3-D plotting programs

Card 4: (2I5)

note	columns	variable	entry
(1)	1 - 5	NOD(1)	Global node number defining the ROTOR of the bearing element (nodal point 1)
(1)	6 - 10	NOD(2)	Global node number defining the STATOR of the bearing element (nodal point 2)

NOTES/

(1) The bearing element is a two node member. One node defines the rotor (shaft) of the engine and the other node defines the stator (housing) of the bearing. During the input of the bearing nodes and their degrees of freedom, the node defining the ROTOR must come first (nodal point 1).

A2.3 Rotating Imbalance Loads

In bearing problems, the following card must be used to input the rotating loads data. It follows ADINA regular concentrated loads input cards.

Concentrated Rotating Loads Data (3I5,2F10.0,I5,2F10.0)

note	columns	variable	entry
(1)	1 - 5	NOD	Node number to which this rotating load is applied; GE.1 and LE. NUMNP
	6 - 10	IDIRN	Degrees of freedom number for this load component: Enter number "2"
	11 - 15	NCUR	Time function number that describes the time dependence f(t) of the load; GE.1 and LE. NLCUR
	16 - 25	FAC	Function multiplier used to scale f(t) to obtain the load at time "t"
	26 - 35	ARTM	Arrival time of the forcing function
	36 - 40	KL	Node number increment used to generate loads at the missing nodes; EQ.0 no generation

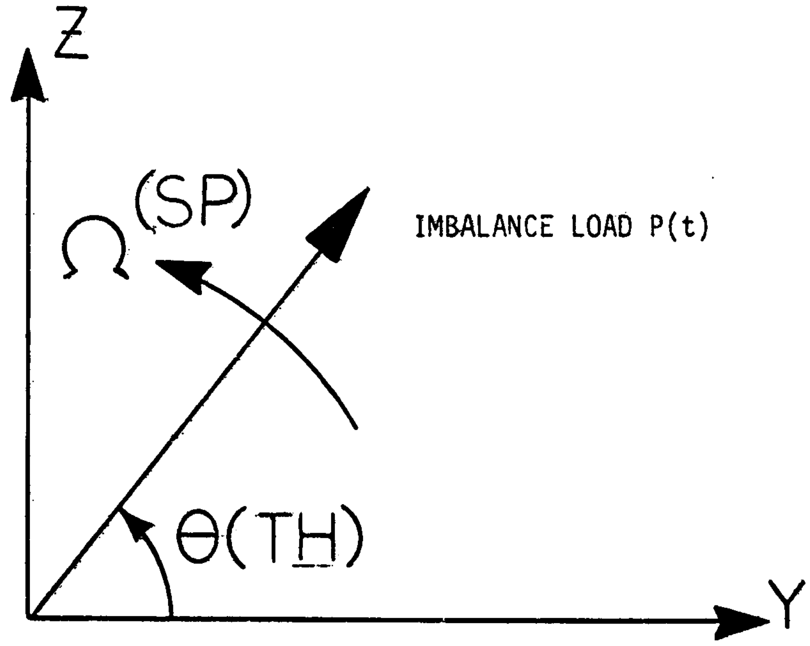
Concentrated Rotating Loads Data (3I5,2F10.0,I5,2F10.0)
continued ...

note	columns	variable	entry
(2)	41 - 50	TH	Initial phase angle of rotating imbalance load applied at node NOD
(2)	51 - 60	SP	Rotating speed of imbalance load applied at node NOD (RAD/SEC)

NOTES/

(1) Input as many cards in this section as the number of nodes where rotating imbalance loads are applied; one card for each node.

(2) The initial phase angle and speed of rotating load measured as shown in the following graph:



A2.4 Plotting Capabilities

During the output phase, the ADINA flow was interrupted to save the necessary bearing responses on a tape to be plotted in a separate plot job. The displacement velocity acceleration and bearing force components of bearing stators and rotors are stored. This is done in subroutine "WRITE". More details on the plotting capabilities are given in Appendices 4 and 5 which list the various plot routines.

APPENDIX 3: LIST OF ROTOR-BEARING-STATOR UPDATES TO ADINA CODE

The listing which follows includes:

(1) Required JCL

(2) Rotor-Bearing-Stator Coding Update

ORIGINAL PAGE IS
OF POOR QUALITY

```
//NASAPR    JOB XXXXX,ZEID
***JOBPARM SKIP=YES,FORMS=1PWB
// EXEC SAVE,DATASET=CIDATA
XXSAVE     PROC TEMP='&&',DATASET=DATASET,PRIME=20,SCNDRY=10,VOL=,
XX              DISP=PASS,STATUS=MOD,UNIT=SYSDA,FILE=1,LABEL=SL,
XX              DEN=4,BLK=6160,SPACE=TRK,RETPD=0
*****************************************************************
***        PURPOSE: PROC TO TAKE INPUT CARDS TO A TEMPORARY DISK FILE *
***        ID: AG1P0170                  MODIFIED: 7 JUL 81 (DEH)    *
*****************************************************************
XXSAVE     EXEC PGM=AG1P0171,REGION=64K
XXSTEPLIB   DD  DSN=ACAD.COMPLIB,DISP=SHR
XX          DD  DSN=SYS2.PL1LIB,DISP=SHR
XX          DD  DSN=SYS2.CMPILERS,DISP=SHR
XXSYSPRINT  DD  SYSOUT=A
XXOUTPUT    DD  DSN=&TEMP.&DATASET,DISP=(&STATUS,&DISP),VOL=SER=&VOL,
XX              LABEL=(&FILE,&LABEL,RETPD=&RETPD),UNIT=&UNIT,
XX              SPACE=(&SPACE,(&PRIME,&SCNDRY),RLSE),
XX              DCB=(RECFM=FB,LRECL=80,BLKSIZE=&BLK,DEN=&DEN)
//SYSIN     DD *                     GENERATED STATEMENT
// EXEC USERPAN,DATASET=BEAR
XXUSERPAN PROC  TEMP='&&',DATASET=DATASET,PRIME=20,SCNDRY=10,DEN=4,
XX              VOL=,LABEL=SL,FILE=1,DISP=PASS,STATUS=MOD,UNIT=SYSDA,
XX              BLK=6160,RETPD=0,SPACE=TRK
*****************************************************************
***       PURPOSE: ALLOWS ACCESS TO USER.PANVALET LIBRARY           *
***       INSTALLER: GARY SPONSELLER      MODIFIED: 7 JUL 81 (DEH)  *
*****************************************************************
XXGO       EXEC PGM=PAN#1,REGION=128K
XXSTEPLIB   DD  DSN=PAN.LOADLIB,DISP=SHR
XXPANDD1    DD  DSN=USER.PANVALET,DISP=SHR
XXPANDD2    DD  DSN=&TEMP.&DATASET,UNIT=&UNIT,DISP=(&STATUS,&DISP),
XX              SPACE=(&SPACE,(&PRIME,&SCNDRY),RLSE),VOL=SER=&VOL,
XX              DCB=(RECFM=FB,LRECL=80,BLKSIZE=&BLK,DEN=&DEN),
XX              LABEL=(&FILE,&LABEL,RETPD=&RETPD)
XXPANDD9    DD  UNIT=SYSDA,SPACE=(CYL,(3,1,1)),
XX              DCB=(RECFM=FB,LRECL=80,BLKSIZE=&BLK,DSORG=PO)
XXSYSPRINT DD   SYSOUT=A
XXSYSPUNCH DD   SYSOUT=B
//SYSIN     DD *                     GENERATED STATEMENT
// EXEC FORT,DATASET=BEAR
XXFORT     PROC  SOURCE=,OBJ=NODECK,MAP=NO,LIST=NO,TEMP='&&',
XX              DATASET=DATASET,ID=,CODE=EBCDIC,LINECNT=60,DISP=PASS
*****************************************************************
***        PURPOSE: COMPILES A FORTRAN PROGRAM (G1 COMPILER).       *
***        MODIFIED: 16 JUL 80 (ELW=W)                              *
*****************************************************************
XXFORT     EXEC  PGM=IGIFORT,REGION=128K,
XX              PARM=('&SOURCE.SOURCE,&OBJ,&MAP.MAP,&LIST.LIST,&ID.ID,',
XX              '&CODE,LINECNT=&LINECNT')
XXSTEPLIB   DD  DSN=SYS2.FTG1COMP,DISP=SHR
XXSYSPRINT  DD  SYSOUT=A
XXSYSPUNCH  DD  SYSOUT=B
XXSYSLIN    DD  DSN=&LOADSET,UNIT=SYSDA,SPACE=(400,(1000,200)),
XX              DCB=BLKSIZE=400,DISP=(MOD,PASS)
XXSYSIN     DD  DSN=&TEMP.&DATASET,DISP=(OLD,&DISP)
// EXEC GOFORTL,LIB=MELIB2,PROG=ADINA,GOSIZE=950K,DISK=
XXGOFORTL  PROC  GOSIZE=128K,EP=MAIN,STATUS=MOD,DISP=PASS,FILE=1,
XX              LIB=DUMMY,PROG=DUMMY,LABEL=SL,UNIT=SYSDA,VOL=,DEN=4,
XX              UNIT4=SAVED,SYSLIB=DUMMY,TEMP='&&',CC=4,BLK=6160,
XX              DATASET=DATASET,MAP=,DISK='DUMMY,',PREC=SP,
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
XX           PRIME=20,SCNDRY=10,PLOT='DUMMY,',OUTPUT=,PREFIX=PLIB
************************************************************************
***          PURPOSE: EXECUTES A FORTRAN OBJECT PROGRAM RESIDING        *
***                   IN A DISK PDS LIBRARY.                            *
***          DATE MODIFIED:  7 JUL 81  (DEH)                            *
************************************************************************
XXGO         EXEC  PGM=LOADER,REGION=&GOSIZE,COND=(&CC,LT),
XX                 PARM='&MAP,MAP,SIZE=&GOSIZE,EP=&EP'
XXSYSLOUT    DD    SYSOUT=A
//SYSLIN DD DSN=&LOADSET,DISP=OLD
X/SYSLIN     DD    DSN=&PREFIX..&LIB(&PROG),DISP=SHR
//           DD DSN=&PREFIX..&LIB.(&PROG),DISP=SHR
XXSYSLIB     DD    DSN=SYS2.FTMILIB,DISP=SHR
XX           DD    DSN=PLIB.&SYSLIB,DISP=SHR
XX           DD    DSN=IMSL.FS300002.&PREC,DISP=SHR
XX           DD    DSN=SYS2.SUBPGM,DISP=SHR
XX           DD    DSN=ACAD.SUBLIB,DISP=SHR
//FT01F001         DD   UNIT=SYSDA,DISP=(NEW,DELETE),SPACE=(CYL,(3,3)),
//                 DCB=(RECFM=VBS,LRECL=3152,BLKSIZE=3156,BUFNO=1)
X/FT01F001   DD    &DISK,UNIT=SYSDA,SPACE=(CYL,(2,1)),
XX                 DCB=(RECFM=VBS,LRECL=1000,BLKSIZE=3156)
//FT02F001         DD   UNIT=SYSDA,DISP=(NEW,DELETE),SPACE=(CYL,(3,3)),
//                 DCB=(RECFM=VBS,LRECL=3152,BLKSIZE=3156,BUFNO=1)
X/FT02F001   DD    &DISK,UNIT=SYSDA,SPACE=(CYL,(2,1)),DCB=*.FT01F001
//FT03F001         DD   UNIT=SYSDA,DISP=(NEW,DELETE),SPACE=(CYL,(1,1)),
//                 DCB=(RECFM=VBS,LRECL=3152,BLKSIZE=3156,BUFNO=1)
X/FT03F001   DD    &DISK,UNIT=SYSDA,SPACE=(CYL,(2,1)),DCB=*.FT01F001
//FT04F001         DD   UNIT=SYSDA,DISP=(NEW,DELETE),SPACE=(CYL,(3,3)),
//                 DCB=(RECFM=VBS,LRECL=3152,BLKSIZE=3156,BUFNO=1)
X/FT04F001   DD    DDNAME=&UNIT4
XXSAVED      DD    DSN=&TEMP.&DATASET,DISP=(&STATUS,&DISP),UNIT=&UNIT,
XX                 SPACE=(TRK,(&PRIME,&SCNDRY),RLSE),VOL=SER=&VOL,
XX                 DCB=(RECFM=FB,LRECL=80,BLKSIZE=&BLK,DEN=&DEN),
XX                 LABEL=(&FILE,&LABEL)
XXWORK       DD    &DISK,UNIT=SYSDA,SPACE=(CYL,(2,1)),DCB=*.FT01F001
//FT05F001         DD   DDNAME=SYSIN
X/FT05F001   DD    DDNAME=SYSIN
//FT06F001         DD   SYSOUT=A
X/FT06F001   DD    SYSOUT=(A,,&OUTPUT),DCB=RECFM=FA
//FT07F001         DD   UNIT=SYSDA,DISP=(NEW,DELETE),SPACE=(CYL,(3,3)),
//                 DCB=(RECFM=VBS,LRECL=3152,BLKSIZE=3156,BUFNO=1)
X/FT07F001   DD    SYSOUT=B
XXPLOTTAPE   DD    &PLOT.SYSOUT=G
//FT08F001         DD   UNIT=SYSDA,DISP=(NEW,DELETE),SPACE=(CYL,(3,3)),
//                 DCB=(RECFM=VBS,LRECL=3152,BLKSIZE=3156,BUFNO=1)
//FT09F001         DD   UNIT=SYSDA,DISP=(NEW,DELETE),SPACE=(CYL,(1,1)),
//                 DCB=(RECFM=VBS,LRECL=3152,BLKSIZE=3156,BUFNO=1)
//FT10F001         DD   UNIT=SYSDA,DISP=(NEW,DELETE),SPACE=(CYL,(3,3)),
//                 DCB=(RECFM=VBS,LRECL=3152,BLKSIZE=3156,BUFNO=1)
//FT11F001         DD   UNIT=SYSDA,DISP=(NEW,DELETE),SPACE=(CYL,(3,3)),
//                 DCB=(RECFM=VBS,LRECL=3152,BLKSIZE=3156,BUFNO=1)
//FT12F001         DD   UNIT=SYSDA,DISP=(NEW,DELETE),SPACE=(CYL,(3,3)),
//                 DCB=(RECFM=VBS,LRECL=3152,BLKSIZE=3156,BUFNO=1)
//FT13F001         DD   UNIT=SYSDA,DISP=(NEW,DELETE),SPACE=(CYL,(3,3)),
//                 DCB=(RECFM=VBS,LRECL=3152,BLKSIZE=3156,BUFNO=1)
//FT20F001 DD DSN=USER.LISTN1,DISP=(NEW,CATLG),
//              SPACE=(TRK,(150,20),RLSE),
//              UNIT=SYSDA,VOL=SER=ACAD01,
//              DCB=(RECFM=FB,BLKSIZE=12960,LRECL=80)
//SYSIN DD DSN=&&CIDATA,DISP=(OLD,DELETE)
```

```
   IEF653I  SUBSTITUTION  JCL  - DSN=&&CIDATA,DISP=(MOD,PASS),VOL=SER=,
   IEF653I  SUBSTITUTION  JCL  - LABEL=(1,SL,RETPD=0),UNIT=SYSDA,
   IEF653I  SUBSTITUTION  JCL  - SPACE=(TRK,(20,10),RLSE),
   IEF653I  SUBSTITUTION  JCL  - DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160,DEN=4)
   IEF653I  SUBSTITUTION  JCL  - DSN=&&BEAR,UNIT=SYSDA,DISP=(MOD,PASS),
   IEF653I  SUBSTITUTION  JCL  - SPACE=(TRK,(20,10),RLSE),VOL=SER=,
   IEF653I  SUBSTITUTION  JCL  - DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160,DEN=4),
   IEF653I  SUBSTITUTION  JCL  - LABEL=(1,SL,RETPD=0)
   IEF653I  SUBSTITUTION  JCL  - DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160,DSORG=PO)
   IEF653I  SUBSTITUTION  JCL  - PARM=('SOURCE,NODECK,NOMAP,NOLIST,ID,',
   IEF653I  SUBSTITUTION  JCL  - 'EBCDIC,LINECNT=60')
   IEF653I  SUBSTITUTION  JCL  - DSN=&&BEAR,DISP=(OLD,PASS)
   IEF653I  SUBSTITUTION  JCL  - PGM=LOADER,REGION=950K,COND=(4,LT),
   IEF653I  SUBSTITUTION  JCL  - PARM='MAP,SIZE=950K,EP=MAIN'
   IEF653I  SUBSTITUTION  JCL  - DSN=PLIB.MELIB2(ADINA),DISP=SHR
   IEF653I  SUBSTITUTION  JCL  - DSN=PLIB.MELIB2(ADINA),DISP=SHR
   IEF653I  SUBSTITUTION  JCL  - DSN=PLIB.DUMMY,DISP=SHR
   IEF653I  SUBSTITUTION  JCL  - DSN=IMSL.FS300002.SP,DISP=SHR
   IEF653I  SUBSTITUTION  JCL  - UNIT=SYSDA,SPACE=(CYL,(2,1)),
   IEF653I  SUBSTITUTION  JCL  - UNIT=SYSDA,SPACE=(CYL,(2,1)),DCB=*.FT01F001
   IEF653I  SUBSTITUTION  JCL  - UNIT=SYSDA,SPACE=(CYL,(2,1)),DCB=*.FT01F001
   IEF653I  SUBSTITUTION  JCL  - DDNAME=SAVED
   IEF653I  SUBSTITUTION  JCL  - DSN=&&DATASET,DISP=(MOD,PASS),UNIT=SYSDA,
   IEF653I  SUBSTITUTION  JCL  - SPACE=(TRK,(20,10),RLSE),VOL=SER=,
   IEF653I  SUBSTITUTION  JCL  - DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160,DEN=4),
   IEF653I  SUBSTITUTION  JCL  - LABEL=(1,SL)
   IEF653I  SUBSTITUTION  JCL  - UNIT=SYSDA,SPACE=(CYL,(2,1)),DCB=*.FT01F001
   IEF653I  SUBSTITUTION  JCL  - SYSOUT=(A,,),DCB=RECFM=EA
   IEF653I  SUBSTITUTION  JCL  - DUMMY,SYSOUT=G
   IEF648I  INVALID DISP FIELD- PASS SUBSTITUTED
ALLOC. FOR NASAPR  SAVE
172   ALLOCATED  TO  STEPLIB
161   ALLOCATED  TO
161   ALLOCATED  TO
161   ALLOCATED  TO  SYS00062
JES2  ALLOCATED  TO  SYSPRINT
26E   ALLOCATED  TO  OUTPUT
JES2  ALLOCATED  TO  SYSIN
NASAPR SAVE - STEP WAS EXECUTED - COND CODE 0000
   ACAD.COMPLIB                                         KEPT
   VOL SER NOS= ACAD02.
   SYS2.PL1LIB                                          KEPT
   VOL SER NOS= SPOOL2.
   SYS2.CMPILERS                                        KEPT
   VOL SER NOS= SPOOL2.
   SYSCTLG.VSPOOL2                                      KEPT
   VOL SER NOS= SPOOL2.
   JES2.JOB00137.SO0103                                 SYSOUT
   SYS81229.T091819.RA000.NASAPR.CIDATA                 PASSED
   VOL SER NOS= WORK03.
   JES2.JOB00137.SI0101                                 SYSIN
STEP /SAVE     / START 81229.0918
STEP /SAVE     / STOP  81229.0918 CPU     0MIN 00.41SEC SRB     0MIN 00.06SEC
ALLOC. FOR NASAPR  GO
269   ALLOCATED  TO  STEPLIB
161   ALLOCATED  TO  SYS00064
172   ALLOCATED  TO  PANDD1
26E   ALLOCATED  TO  PANDD2
166   ALLOCATED  TO  PANDD9
```

```
IEF237I JES2 ALLOCATED TO SYSPRINT
IEF237I JES2 ALLOCATED TO SYSPUNCH
IEF237I JES2 ALLOCATED TO SYSIN
IEF142I NASAPR GO - STEP WAS EXECUTED - COND CODE 0000
IEF285I   PAN.LOADLIB                                         KEPT
IEF285I   VOL SER NOS= STOR02.
IEF285I   SYSCTLG.VSPOOL2                                     KEPT
IEF285I   VOL SER NOS= SPOOL2.
IEF285I   USER.PANVALET                                       KEPT
IEF285I   VOL SER NOS= ACA002.
IEF285I   SYS81229.T091819.RA000.NASAPR.BEAR                  PASSED
IEF285I   VOL SER NOS= WORK03.
IEF285I   SYS81229.T091819.RA000.NASAPR.R0000001              DELETED
IEF285I   VOL SER NOS= WORK01.
IEF285I   JES2.JOB00137.SO0104                                SYSOUT
IEF285I   JES2.JOB00137.SO0105                                SYSOUT
IEF285I   JES2.JOB00137.SI0102                                SYSIN
IEF373I STEP /GO      / START 81229.0918
IEF374I STEP /GO      / STOP  81229.0918 CPU    0MIN 08.37SEC SRB
IEF236I ALLOC. FOR NASAPR FORT
IEF237I 162  ALLOCATED TO STEPLIB
IEF237I 161  ALLOCATED TO SYS00066
IEF237I JES2 ALLOCATED TO SYSPRINT
IEF237I JES2 ALLOCATED TO SYSPUNCH
IEF237I 166  ALLOCATED TO SYSLIN
IEF237I 26E  ALLOCATED TO SYSIN
IEF142I NASAPR FORT - STEP WAS EXECUTED - COND CODE 0000
IEF285I   SYS2.FTG1COMP                                       KEPT
IEF285I   VOL SER NOS= STOR03.
IEF285I   SYSCTLG.VSPOOL2                                     KEPT
IEF285I   VOL SER NOS= SPOOL2.
IEF285I   JES2.JOB00137.SO0106                                SYSOUT
IEF285I   JES2.JOB00137.SO0107                                SYSOUT
IEF285I   SYS81229.T091819.RA000.NASAPR.LOADSET               PASSED
IEF285I   VOL SER NOS= WORK01.
IEF285I   SYS81229.T091819.RA000.NASAPR.BEAR                  PASSED
IEF285I   VOL SER NOS= WORK03.
IEF373I STEP /FORT    / START 81229.0918
IEF374I STEP /FORT    / STOP  81229.0922 CPU    1MIN 10.07SEC SRB
IEF236I ALLOC. FOR NASAPR GO
IEF237I JES2 ALLOCATED TO SYSLOUT
IEF237I 166  ALLOCATED TO SYSLIN
IEF237I 164  ALLOCATED TO
IEF237I 161  ALLOCATED TO SYS00068
IEF237I 162  ALLOCATED TO SYSLIB
IEF237I 164  ALLOCATED TO
IEF237I 172  ALLOCATED TO
IEF237I 269  ALLOCATED TO
IEF237I 172  ALLOCATED TO
IEF237I 166  ALLOCATED TO FT01F001
IEF237I 26E  ALLOCATED TO FT02F001
IEF237I 26E  ALLOCATED TO FT03F001
IEF237I 166  ALLOCATED TO FT04F001
IEF237I 166  ALLOCATED TO SAVED
IEF237I 26E  ALLOCATED TO WORK
IEF237I 26E  ALLOCATED TO FT05F001
IEF237I JES2 ALLOCATED TO FT06F001
IEF237I 26E  ALLOCATED TO FT07F001
IEF237I DMY  ALLOCATED TO PLOTTAPE
IEF237I 166  ALLOCATED TO FT08F001
IEF237I 166  ALLOCATED TO FT09F001
```

```
IEF237I 166   ALLOCATED TO FT10F001
IEF237I 26E   ALLOCATED TO FT11F001
IEF237I 26E   ALLOCATED TO FT12F001
IEF237I 166   ALLOCATED TO FT13F001
IEF237I 164   ALLOCATED TO FT20F001
IEF142I NASAPR GO - STEP WAS EXECUTED - COND CODE 0000
IEF285I   JES2.JOB00137.S00108                              SYSOUT
IEF285I   SYS81229.T091819.RA000.NASAPR.LOADSET             PASSED
IEF285I   VOL SER NOS= WORK01.
IEF285I   PLIB.MELIB2                                       KEPT
IEF285I   VOL SER NOS= ACAD01.
IEF285I   SYSCTLG.VSPOOL2                                   KEPT
IEF285I   VOL SER NOS= SPOOL2.
IEF285I   SYS2.FTMILIB                                      KEPT
IEF285I   VOL SER NOS= STOR03.
IEF285I   PLIB.DUMMY                                        KEPT
IEF285I   VOL SER NOS= ACAD01.
IEF285I   IMSL.FS3D0002.SP                                  KEPT
IEF285I   VOL SER NOS= ACAD02.
IEF285I   SYS2.SUBPGM                                       KEPT
IEF285I   VOL SER NOS= STOR02.
IEF285I   ACAD.SUBLIB                                       KEPT
IEF285I   VOL SER NOS= ACAD02.
IEF285I   SYS81229.T091819.RA000.NASAPR.R0000002            DELETED
IEF285I   VOL SER NOS= WORK01.
IEF285I   SYS81229.T091819.RA000.NASAPR.R0000003            DELETED
IEF285I   VOL SER NOS= WORK03.
IEF285I   SYS81229.T091819.RA000.NASAPR.R0000004            DELETED
IEF285I   VOL SER NOS= WORK03.
IEF285I   SYS81229.T091819.RA000.NASAPR.R0000005            DELETED
IEF285I   VOL SER NOS= WORK01.
IEF285I   SYS81229.T091819.RA000.NASAPR.DATASET             PASSED
IEF285I   VOL SER NOS= WORK01.
IEF285I   SYS81229.T091819.RA000.NASAPR.R0000006            DELETED
IEF285I   VOL SER NOS= WORK03.
IEF285I   SYS81229.T091819.RA000.NASAPR.CIDATA              DELETED
IEF285I   VOL SER NOS= WORK03.
IEF285I   JES2.JOB00137.S00109                              SYSOUT
IEF285I   SYS81229.T091819.RA000.NASAPR.R0000007            DELETED
IEF285I   VOL SER NOS= WORK03.
IEF285I   SYS81229.T091819.RA000.NASAPR.R0000008            DELETED
IEF285I   VOL SER NOS= WORK01.
IEF285I   SYS81229.T091819.RA000.NASAPR.R0000009            DELETED
IEF285I   VOL SER NOS= WORK01.
IEF285I   SYS81229.T091819.RA000.NASAPR.R0000010            DELETED
IEF285I   VOL SER NOS= WORK01.
IEF285I   SYS81229.T091819.RA000.NASAPR.R0000011            DELETED
IEF285I   VOL SER NOS= WORK03.
IEF285I   SYS81229.T091819.RA000.NASAPR.R0000012            DELETED
IEF285I   VOL SER NOS= WORK03.
IEF285I   SYS81229.T091819.RA000.NASAPR.R0000013            DELETED
IEF285I   VOL SER NOS= WORK01.
IEF285I   USER.LISTN1                                       CATALOGED
IEF285I   VOL SER NOS= ACAD01.
IEF373I STEP /GO      / START 81229.0922
IEF374I STEP /GO      / STOP  81229.0923 CPU -    0MIN 20.85SEC SRB
IEF237I 26E  ALLOCATED TO SYS00001
IEF285I   SYS81229.T092345.RA000.NASAPR.R0000001            KEPT
IEF285I   VOL SER NOS= WORK03.
IEF285I   SYS81229.T091819.RA000.NASAPR.BEAR                DELETED
IEF285I   VOL SER NOS= WORK03.
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
IEF237I 166  ALLOCATED TO SYS00003
IEF285I   SYS81229.T092345.RA000.NASAPR.R0000003        KEPT
IEF285I   VOL SER NOS= WORK01.
IEF285I   SYS81229.T091819.RA000.NASAPR.LOADSET         DELETED
IEF285I   VOL SER NOS= WORK01.
IEF237I 166  ALLOCATED TO SYS00005
IEF285I   SYS81229.T092345.RA000.NASAPR.R0000005        KEPT
IEF285I   VOL SER NOS= WORK01.
IEF285I   SYS81229.T091819.RA000.NASAPR.DATASET         DELETED
IEF285I   VOL SER NOS= WORK01.
IEF375I  JOB /NASAPR  / START 81229.0918
IEF376I  JOB /NASAPR  / STOP  81229.0923 CPU    1MIN 39.70SEC SRB
```

```
                    MAIN                      DATE = 81229          09/18/34
C             DATA SET AAMAIN    AT LEVEL TMP AS OF 08/17/81
C *CDC* *DECK OVL00
C *CDC*        OVERLAY (ADINA,0,0)
C *CDC* *DECK ADINA
C *UNI* )FOR,IS  N.ADINA,  R.ADINA
C *CDC* PROGRAM ADINA (INPUT, OUTPUT , PUNCH, TAPE5=INPUT, TAPE6 =
C *CDC* 1                  OUTPUT, TAPE1, TAPE2, TAPE3, TAPE4,
C *CDC* 2                  TAPE7, TAPE8, TAPE9, TAPE10, TAPE11 ,
C *CDC* 3                  TAPE12, TAPE13, TAPE24, TAPE59, TAPE60, TAPE61,
C *CDC* 4                  TAPE62, TAPE63)
C
C
C
C
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .
C .   T A P E   A L L O C A T I O N                                      .
C .                                                                      .
C .        . TAPE 1   STORES LINEAR ELEMENT GROUP DATA                   .
C .                   ( WHEN NEGL.NE.0 ONLY )                            .
C .                                                                      .
C .        . TAPE 2   STORES NONLINEAR ELEMENT GROUP DATA                .
C .                                                                      .
C .        . TAPE 3   STORES EXTERNALLY APPLIED LOADS                    .
C .                                                                      .
C .        . TAPE 4   STORES THE LINEAR STIFFNESS MATRIX                 .
C .                   ( WHEN NEGL.NE.0 ONLY )                            .
C .                                                                      .
C .        . TAPES 5,6 ARE INPUT,OUTPUT TAPES                            .
C .                                                                      .
C .        . TAPE 7   STORES EFFECTIVE LINEAR COEFFICIENT MATRIX         .
C .                                                                      .
C .        . TAPE 8   STORES SEQUENTIALLY                                .
C .            (1) ID ARRAY                                              .
C .            (2) DURING INPUT                                          .
C .                  INITIAL DISP, VEL, ACC VECTORS                      .
C .                ON OUTPUT                                             .
C .                  FINAL DISP, VEL, ACC VECTORS                        .
C .                  AND NONLINEAR ELEMENT GROUP DATA FOR RESTART        .
C .                                                                      .
C .        . TAPE 9   STORES NODAL COORDINATES FOR PRESSURE LOAD         .
C .                   CALCULATIONS                                       .
C .                                                                      .
C .        . TAPE 10 STORES                                              .
C .                 D - L FACTORS OF EFFECTIVE LINEAR OR NONLINEAR       .
C .                 STIFFNESS MATRIX IN TIME INTEGRATION                 .
C .                                                                      .
C .        . TAPE 11 STORES SEQUENTIALLY                                 .
C .            (1) CONSISTENT MASS MATRIX (IF IMASS.EQ.2)                .
C .            (2) DAMPING VECTOR                                        .
C .                                                                      .
C .        . TAPE 12 STORES THE EFFECTIVE NONLINEAR STIFFNESS MATRIX     .
C .            IN THE TIME INTEGRATION - OR -                            .
C .            THE NONLINEAR STIFFNESS MATRIX FOR EIGENSYSTEM SOLUTION   .
C .                                                                      .
C .        . TAPE 13 STORES MODE SHAPES AND CIRCULAR FREQUENCIES         .
C .            (IF FREQUENCIES SOLUTION WAS REQUESTED)                   .
C .                                                                      .
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
C *   * TAPE 24 STORES NODAL POINT TEMPERATURES                               *
C *                                                                           *
C *   * TAPE 59 STORES PREPROCESSED INPUT DATA (IF USED)                      *
C *                                                                           *
C *   * TAPES 60 TO 63 ARE PORTHOLE FILES FOR SAVING NODAL/ELEMENT            *
C *     RESPONSES (IF REQUESTED)                                              *
C *                                                                           *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /SOL/ NUMNP,NEQ,NWK,NWM,NWC,NUMEST,MIDEST,MAXEST,NSTE,MA
      COMMON /DIM/ N0,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /DIMEL/ N101,N102,N103,N104,N105,N106,N107,N108,N109,N110,
     1               N111,N112,N113,N114,N120,N121,N122,N123,N124,N125
      COMMON /EL/ IND,ICOUNT,NPAR(20),NUMEG,NEGL,NEGNL,IMASS,IDAMP,ISTAT,
     1          NDOF,KLIN,IEIG,IMASSN,IDAMPN
      COMMON /CONST/ DT,DTA,A0,A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11
     1              ,A12,A13,A14,A15,A16,A17,A18,A19,A20,IOPE
      COMMON /LOA/ TEND,NTFN,NPTM,NLOAD,IDGRAV,NPR2,NPR3,NODE3
      COMMON /JUNK/ HED(12),MTOT,LPROG
      COMMON /VAR/ NG,MODEX,IUPDT,KSTEP,ITEMAX,IEQREF,ITE,KPRI,
     1           IREF,IEQUIT,IPRI,KPLOTN,KPLOTE
      COMMON /NORMS/ RNORM, RENORM, RTOL,DNORMM
      COMMON /PRCON/ IDATWR,IPRIC,NPB,IDC,IVC,IAC,IPC,IPNODE(3,15)
      COMMON /ADINAI/ OPVAR(7),TSTART,IRINT,ISTOTE
      COMMON /DPR/ ITWO
      COMMON /GAUSS/ XG(4,4),WGT(4,4),EVAL2(9,2),EVAL3(27,3),E1,E2,E3
      COMMON /ELGLTH/ NFIRST,NLAST,NBCEL
      COMMON /ELSTP/ TIME,IDTHF
      COMMON /ADDB/ NEQL,NEQR,MLA,NBLOCK
      COMMON /RANDI/ N0A,N10,IELCPL
      COMMON /FREQIF/ ISTOH,N1A,N1B,N1C
      COMMON /TEMPRT/ TEMP1,TEMP2,ITEMPR,ITP96,N6A,N6B
      COMMON /MDFRDM/ IDOF(6)
      COMMON /BLOCKS/ NSREFB,NEQITB,NPRIB,NODSVB,LEMSVB,ISREFB(3,10),
     1               IEQITB(3,10),IPRIB(3,10),INODB(3,10),IELMB(3,10)
      COMMON /PORT/ INPORT,JNPORT,NPUTSV,LUNODE,LU1,LU2,LU3,JDC,JVC,JAC
      COMMON /RANDAC/ NR(2),LR(2)
      COMMON /BEAR/ IBEAR, MTOTB
C     ESTABLISH A DYNAMICALLY ALLOCATED ARRAY B TO BE
C     USED FOR BEARING CALCULTIONS.
      COMMON /BEAR2/ B(5000)
      COMMON A(400010)
      INTEGER IA(1)
      REAL A
      REAL B
      EQUIVALENCE (A(1),IA(1))
C
C
C
C
C    NOTE **- CURRENT IBM VERSION OF THE PROGRAM CAN BE USED ONLY WITH A
C         MAX. OF 300000 ELEMENTS IN THE STIFFNESS MATRIX.
C
C         RANDOM ACCESS I/O IS USED IN THE FOLLOWING SUBROUTINES ALSO -
C         * ELCAL, ASSEM, LOADEF, COLSOL, EQUIT, STRESS, RSTART.
```

C - 2

```
                    MAIN                   DATE = 81229          09/18/44
C      RUSS.-TDFE. THDFE. SMEL. BANDET. SAVARC  *
C
C         PRIOR AND AFTER A RANDOM ACCESS READ/WRITE THE FOLLOWING CARDS
C      HAVE BEEN INCLUDED
C
C         * * * * * *       R A N D O M   A C C E S S         * * * *
C
C
C      CREATE RANDOM FILES 2,10 WITH NUMBER INDEX
C
C         * * * * * *       R A N D O M   A C C E S S         * * * *
C
       NR(1)=190
       NR(2)=190
       LR(1)=3000
       LR(2)=3000
       DEFINE FILE 10 (190,3000,U,NREC10)
       DEFINE FILE  2 (190,3000,U,NREC2)
C
C         * * * * * *       R A N D O M   A C C E S S         * * * *
C
C
C
       MTOT=40000
       MTOT3 = 5000
C *CDC*          LTWO=1
       ITWO=2
       NBCST=MTOT
        NBCST=1
C
C
  200  NUMEST=0
       MAXEST=0
       NBCEL=0
       NREAD=2
       NUMREF=0
       ITE=0
       KPRI=1
       KSTEP=0
       IND=0
       ICOUNT=2
C
C
C      I N P U T   P H A S E
C
C
       CALL SECOND (TIM1)
C
       NO=1 + NBCST
C
C      STORAGE IS ALLOCATED IN ADINI BECAUSE THE REQUIRED
C      VARIABLES ARE NOT KNOWN YET
C
C
C *CDC*          CALL OVERLAY (SHADINA,1,0,6HRECALL)
       CALL ADINI
C
C
```

```
C
C     O B T A I N   E L E M E N T   I N F O R M A T I O N S
C
C     CLEAR ARRAY FOR CALCULATION OF COLUMN HEIGHTS
C
      NN=N5 + NEQ
      DO 2 I=N5,NN
    2 IA(I)=0
      N6=N5 + NEQ
C
C     INITIALIZE TEMPERATURE ARRAY
C
      IF (ITP96.EQ.0) GO TO 14
      N6A=N5 + NEQ + 1
      N6B=N6A
      N6=N6A + (NUMNP+1)*ITWO
      NN=N6 - 1
      READ (24) (A(I),I=N6A,NN)
      CALL TCHECK (A(N6A),TSTART)
      BACKSPACE 24
C
   14 CALL ELCAL
C
      CALL SECOND (TIM2)
C
C     STORE MAXA ARRAY
C
      IF (IOPE.NE.3) CALL ADDRES (A(N1),A(N5))
C
C
C     S T O R A G E   C A L C U L A T I O N S
C
C
C     TEST FOR AVAILABILITY OF HIGH SPEED STORAGE AND CALCULATE
C     MAXIMUM BLOCKSIZE, NUMBER OF BLOCKS, AND BLOCK COUPLING
C
C       1. STORAGE FOR LOAD VECTOR CALCULATIONS
C
      MSTORE=(NEQ+1) + NDOF*NUMNP + (NEQ + 2*NPTM + NTFN*NSTE + 2*NLOAD)
     1       *ITWO + 4*NLOAD + NTFN*ITWO
      IF (IDGRAV.EQ.1) MSTORE=MSTORE + 3*ITWO + NEQ + NEQ*ITWO
      WRITE (6,2200)
      CALL SIZE(MSTORE)
C
C       2. STORAGE FOR MATRIX ASSEMBLAGE PHASE AND TIME INTEGRATION
C
C     CENTRAL DIFFERENCE MATHOD
C
      IF (IOPE.NE.3) GO TO 5
      ISV=(IVC + JVC + 1)/2
      ISA=(IAC + JAC + 1)/2
      MSTORE=NEGNE + (3*ISV+ISA)*NEQ*ITWO + ITEMPR*(NUMNP + 1)*ITWO
      MTEMP=NDOF*NUMNP
      IF (MTEMP.LT.NEQ*ITWO) MTEMP=NEQ*ITWO
      MSTORE=MSTORE + MTEMP + MAXEST + NBCEL
      WRITE (6,2202)
      CALL SIZE (MSTORE)
      NBLOCK=1
```

```
      ISTOH=0
      GO TO 45
C
C     STATIC ANALYSIS AND IMPLICIT TIME INTEGRATION
C
    5 MSTORE=(NEQ + 1) + 3*NEQ*ITWO + ITEMPR*(NUMNP + 1)*ITWO + MAXEST
     1      + NBCEL
      MTEMP=2*NEQ*ITWO
      IF (MTEMP.LT.(NDOF*NUMNP)) MTEMP=NDOF*NUMNP
      MSTORE=MSTORE + MTEMP
      IF (ISTAT.EQ.1) MSTORE=MSTORE + 2*NEQ*ITWO
      IF (IMASS.EQ.1) MSTORE=MSTORE + NEQ*ITWO
      N1A=N1 + NEQ + 1
      N1B=N1A + NEQ
      IBLOCK=4
      NBLOCK=1
   10 MELST=IBLOCK*NEGNL + 2*IBLOCK
      MELST=MELST + (IEIG + 1)*IBLOCK + 1
      ISTORL=(MTOT - MSTORE - MELST)/ITWO
      IF (ISTOTE.GT.0) ISTORL=ISTOTE
      IF (ISTORL.GT.0) GO TO 15
      WRITE (6,2203)
      STOP
C
   15 CALL SBLOCK (A(N1),A(N1A),A(N1B),ISTORL,NBLOCK,NEQ,NWK,ISTOH)
C
      IF (ISTOTE.GT.0) GO TO 20
      IF (NBLOCK.LE.IBLOCK) GO TO 20
      IBLOCK=IBLOCK*2
      IF (IBLOCK.LT.1000) GO TO 10
      WRITE (6,2204)
      STOP
   20 MAM=NWK/NEQ + 1
C
C        3. SPECIAL CASE   IF ONE BLOCK CASE AND CONSISTENT MASS MATRIX
C                                                          IS USED
C
      IF (IMASS.NE.2 .OR. NBLOCK.GT.1) GO TO 30
      MM=2*ISTOH
      IF (MM.LE.ISTORL) GO TO 30
      NBLOCK=2
      CALL SBLOCK (A(N1),A(N1A),A(N1B),ISTORL,NBLOCK,NEQ,NWK,ISTOH)
C
C        4. STORAGE FOR FREQUENCY ANALYSIS
C
   30 IF (IEIG.EQ.0) GO TO 36
      MSTORE=9*NEQ*ITWO + NEQ + 1000
      IF (IMASS.EQ.1) MSTORE=MSTORE + NEQ*ITWO
      MM=(MTOT - MSTORE)/ITWO
      IF (NBLOCK.GT.1 .OR. IMASS.EQ.2) MM=MM/2
      IF (MM.GE.ISTOH) GO TO 36
      IF (NBLOCK.GT.1 .OR. IMASS.EQ.2) MM=2*MM
      NBLOCK=2
      CALL SBLOCK (A(N1),A(N1A),A(N1B),MM,NBLOCK,NEQ,NWK,ISTOH)
C
C       WRITE TOTAL SYSTEM DATA
C
   36 WRITE (6,2210) NEQ,NWK,MA,MAM,ISTOH,NBLOCK,MTOT
```

```
      WRITE (6,2220)
      NN=N1A + NBLOCK - 1
      WRITE (6,2230) (I,I=1,NBLOCK)
      WRITE (6,2240) (IA(I),I=N1A,NN)
      NN=N1B + NBLOCK - 1
      WRITE (6,2250) (IA(I),I=N1B,NN)
      NN=N1A + NBLOCK
      DO 40 I=1,NBLOCK
   40 IA(NN+I-1)=IA(N1B+I-1)
C
C
C     A S S E M B L A G E   O F   L I N E A R   M A T R I C E S
C
C
      N1B=NN
      N1C=N1B + NBLOCK
      N1D=N1C + NBLOCK*NEGNL
      IF (NBLOCK.EQ.1) N1D=N1C
      N2=N1D + (IEIG + 1)*NBLOCK + 1
      N3=N2 + ISTOH*ITWO
      N4=N3 + ISTOH*ITWO
      IF (NBLOCK.EQ.1 .AND. IMASS.LT.2) N4=N3
   45 IF (IOPE.EQ.3) N4=N1
      N5=N4 + NEQ*ITWO
      N6=N5 + NEQ*ITWO
      IF (IMASS.EQ.0) N6=N4
      IF (ISTAT.EQ.0 .AND. IDGRAV.EQ.1) N6=N5
      N7=N6 + MAXEST + NBCEL
      WRITE (6,2260)
      CALL SIZE(N7)
      CALL SECOND(TIM3)
      IF (MODEX.GT.0) GO TO 50
      IND=2
C
C     CREATE RANDOM ACCESS FILE 10 WITH ASSOCIATED RECORD NUMBER INDEX
C
C       * * * * *       R A N D O M   A C C E S S        * * * *
C
   50 NBLOC1=(IEIG + 1)*NBLOCK + 1
C *CDC*         IF (IOPE.NE.3) CALL OPENMS (10,IA(N1D),NBLOC1,0)
C
C       * * * * *       R A N D O M   A C C E S S        * * * *
C
      IF (MODEX.EQ.0) GO TO 60
      IND=1
      IREF=0
      CALL ASSEM (A(N1),A(N2),A(N3),A(N4),A(N5),A(N6),A(N1A),A(N4),
     1            A(N1C),ISTOH,NBLOCK)
C
C
C     C A L C U L A T E   A N D   S T O R E   L O A D   V E C T O R S
C
C
C *CDC*    60 CALL OVERLAY (5HADINA,1,0,5HRECALL)
   60 CALL ADINI
C
C     CALCULATE STORAGE LOCATIONS FOR TIME INTEGRATION
C
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
      CALL SECOND(TIM4)
      IF (IOPE.EQ.3) N2=N1 + NEQ*ITWO
      N3=N2 + NEQ*ITWO
      N4=N3 + NEQ*ITWO
      N4A=N4 + ISTOH*ITWO
      N4B=N4A + ISTOH*ITWO
      IF (NBLOCK.EQ.1) N4B=N4A
      N5=N4B + NEQ*ITWO
      IF (IOPE.EQ.3) N5=N4
      N6=N5 + NEQ*ITWO
      N6A=N6 + NEQ*ITWO
      NN1=NDOF*NUMNP
      NN2=2*NEQ*ITWO
      IF (NN1.GT.NN2) N6A=N5 + NN1
      IF (IOPE.EQ.3) N6=N5
      NN2=NEQ*ITWO
      IF (NN2.GT.NN1) NN1=NN2
      IF (IOPE.EQ.3) N6A=N6 + NN1
      N6B=N6A + (ITEMPR-1)*(NUMNP+1)*ITWO
      N7=N6B + (NUMNP+1)*ITWO
      N8=N7 + NEQ*ITWO
      IF (IOPE.EQ.3 .AND. ISV.EQ.0) N8=N7
      N9=N8 + NEQ*ITWO
      N10=N9 + NEQ*ITWO
      IF (IOPE.EQ.3) N10=N7 + (ISV + ISA)*NEQ*ITWO
      IF (IMASS.EQ.2) N10=N9
      IF (ISTAT.EQ.0) N10=N7
      N11=N10 + MAXEST + NBCEL
      WRITE (6,2280)
      CALL SIZE (N11)
C
C     INITIALIZE TEMPERATURE ARRAY
C
      IF (ITEMPR.EQ.0) GO TO 64
      NN=N6A + (NUMNP+1)*ITWO - 1
      READ (24) (A(I),I=N6A,NN)
C
C     WRITE INITIALIZED DISPLACEMENTS, VELOCITIES, AND ACCELERATIONS
C        ( OR STARTING DISPL/VEL/ACC IF THIS IS A RESTART JOB )
C
   64 CALL WRITE (A(N1),A(N2),A(N7),A(N8),A(N5),NEQ,NDOF,1)
C
C     IF THIS IS A RESTART JOB TRANSFER NONLINEAR ELEMENT GROUP
C     DATA TO TAPE 2
C
      IF (MODEX.NE.2) GO TO 82
      CALL RSTART (A(N1),A(N2),A(N7),A(N8),A(N10),NEQ,IOPE,2)
C
C
C
C     F R E Q U E N C Y   S O L U T I O N
C
C
   82 CALL SECOND (TIM5)
      IF (IEIG.EQ.3) GO TO 69
      IND=3
      TIME=TSTART + DT
      CALL ASSEM (A(N1),A(N4),A(N4A),A(N2),A(N3),A(N10),A(N1A),A(N6B),
     1            A(N1C),ISTOH,NBLOCK)
```

```
  69 CALL SECOND (TIM6)
C
C
C     T I M E   I N T E G R A T I O N
C
C
      TSUM1=0.
      TSUM2=0.
      TSUM3=0.
      TSUM4=0.
      TSUM5=0.
      TSUM6=0.
      CALL SECOND (TIM7)
      TIM8=TIM7
      IF (MODEX.GT.0) GO TO 88
      WRITE(6,2030)
      GO TO 190
   88 IF (NSTE.EQ.0) GO TO 190
C
      TIME=TSTART
      TIMEP=TSTART
      REWIND 3
      IND=4
      KRINT=0
      NUMP1=(NUMNP+1)*IT#0
C
C     IN CASE OF LINEAR ANALYSIS TRIANGULARIZE EFFECTIVE LINEAR
C     STIFFNESS MATRIX (THE TRIANGULAR FACTORS REMAIN IN CORE
C     PROVIDED THAT
C                 1.   LINEAR ANALYSIS
C                 2.   ONE BLOCK CASE
C                 3.   IMPLICIT TIME INTEGRATION SCHEME IS USED)
C
      CALL SECOND (TIM7)
      IF (KLIN.GT.0 .OR. IOPE.EQ.3) GO TO 94
      NTAPE=4
      IF (ISTAT.EQ.1) NTAPE=7
      CALL COLSOL (A(N1),A(N1A),A(N1B),A(N4),A(N4A),A(N4B),A(NJ),
     1             NEQ,NBLOCK,ISTOH,NTAPE,10,1)
   94 CALL SECOND (TIM8)
      IF (IOPE.EQ.3) GO TO 100
C
C     LUMPED MASS MATRIX IS TAKEN INTO CORE AND NODAL DAMPING VECTOR
C     IS STORED AS FIRST RECORD (IMPLICIT TIME INTEGRATION).
C
      IF (IMASS.NE.1) GO TO 100
      REWIND 11
      NN=N10 - 1
      READ (11) (A(I),I=N9,NN)
      NN=N6 + NEQ*ITWO - 1
      READ (11) (A(I),I=N6,NN)
      REWIND 11
      WRITE (11) (A(I),I=N6,NN)
C
C
C     T I M E   S T E P   I N C R E M E N T A T I O N
C
C        KSTEP .EQ. STEP COUNTER
```

```
C         TIME .EQ. TIME AT WHICH SOLUTION IS REQUIRED
C
C
  100 KSTEP=KSTEP + 1
      TIMEP=TIME + DTA
      TIME=TIME + DT
C
C     STIFFNESS REFORMATION FLAG
C        IREF.EQ.0 IF STIFFNESS IS TO BE REFORMED.
C
      CALL BLKCNT(KSTEP,NSREFB,IREF,ISREFB,NSTE,1)
      IF (IJRE.NE.3 .AND. KSTEP.EQ.1) IREF=0
C
C     FLAG FOR EQUILIBRIUM ITERATION
C        IEQUIT.EQ.0 IF ITERATION IS TO BE PERFORMED
C        IEQUIT.GT.0 IF NO ITERATION IS TO BE PERFORMED
C
      CALL BLKCNT(KSTEP,NEQITB,IEQUIT,IEQITB,NSTE,2) .
C
C     FLAG FOR TRIANGULARIZATION AND/OR SIMPLE REDUCTION AND
C     BACKSUBSTITUTION IN COLSOL
C        KTR.EQ.1    FOR TRIANGULARIZATION PLUS SOLUTION
C        KTR.EQ.2    FOR VECTOR SOLUTION ONLY
C
      KTR=1
      IF (IREF.NE.0) KTR=2
      IF (KLIN.EQ.0) KTR=2
C
C     NEQREF IS THE NUMBER OF TIMES THE NONLINEAR STIFFNESS MATRIX
C     WAS REFORMED
C
      NEQREF=0
  140 REWIND 4
      REWIND 7
C
C     FLAG TO INDICATE CONVERGENCE IN EQUILIBRIUM ITERATION
C        IEQREF.EQ.0    CONVERGENCE
C        IEQREF.EQ.1    NORM OF OUT-OF-BALANCE LOADS IS LARGER THAN NORM
C                       OF INCREMENTAL LOADS (SEE EQUIT)
C
      IEQREF=0
C
C
C     S O L U T I O N   O F   I N C R E M E N T A L   E Q U A T I O N S
C
C
C     CALCULATE LINEAR EFFECTIVE LOADS BALANCED IN CURRENT CONFIGURATION
C
      CALL SECOND (TIM9)
C
      CALL LOADEF (A(N1),A(N1A),A(N2),A(N1),A(N7),A(N8),A(N3),A(N6),
     1             A(N4),A(N9),NBLOCK,ISTOH)
C
      CALL SECOND (TIM10)
C
C     CALCULATE EFFECTIVE NONLINEAR MATRIX AND FINAL EFFECTIVE LOADS
C
      CALL ASSEM (A(N1),A(N4),A(N4A),A(N2),A(N3),A(N10),A(N1A),A(N6B),
```

```
     1            A(N1C),ISTOH,NBLOCK)
C
      CALL SECOND (TIM11)
      IF (KSTEP.EQ.1 .AND. IOPE.NE.3) WRITE (6,2300) TIM11
C
C     SOLVE FOR INCREMENT IN DISPLACEMENT VECTOR. NO TRIANGULARIZATION
C     IF SIMPLE EQUILIBRIUM ITERATION IS TO BE PERFORMED
C
      IF (IOPE.NE.3) CALL COLSOL (A(N1),A(N1A),A(N1B),A(N4),A(N4A),
     1          A(N4B),A(N3),NEQ,NBLOCK,ISTOH,12,10,KTR)
C
      CALL SECOND (TIM12)
      IF (KSTEP.EQ.1 .AND. IOPE.NE.3) WRITE (6,2310) TIM12
      TSUM1=TSUM1 + (TIM10 - TIM9)
      TSUM2=TSUM2 + (TIM11 - TIM10)
      TSUM3=TSUM3 + (TIM12 - TIM11)
C
C
C     I T E R A T I O N   F O R   D Y N A M I C   E Q U I L I B R I U M
C
C
C     NO ITERATION IN LINEAR ANALYSIS
C
      IF (KLIN.EQ.0) GO TO 110
C
      IF (IEQUIT.NE.0) GO TO 110
C
      CALL SECOND (TIM13)
C
      CALL EQUIT (A(N4),A(N3),A(N5),A(N2),A(N7),A(N8),A(N1),A(N6),
     1            A(N9),A(N10),A(N4A),A(N4B),A(N1A),A(N1B),ISTOH)
C
C     IF NO CONVERGENCE IN ITERATION PROCEED TO NEXT DATA CASE
C
      IF (ITE.GT.ITEMAX) GO TO 190
C
      CALL SECOND (TIM14)
      TSUM4=TSUM4 + (TIM14 - TIM13)
C
C     CHECK FOR NO CONVERGENCE IN EQUILIBRIUM ITERATION AND
C     POSSIBLE REFORMATION OF STIFFNESS
C
      IF (IEQREF.EQ.0) GO TO 110
      KTR=1
      IREF=0
      NEQREF=NEQREF + 1
      IF (NUMREF.EQ.0) GO TO 112
      IF (NEQREF.LE.1) GO TO 140
  112 WRITE(6,2040)
      GO TO 190
C
C
C     CALCULATE NEW DISP. VEL. ACC VECTORS AT TIME=TSTART + KSTEP*DT
C     FOR STATIC ANALYSIS AND IMPLICIT TIME INTEGRATION AND ALSO DISP
C     VECTOR AT TIME=TSTART + (KSTEP + 1)*DT FOR CENTRAL DIFFERENCE
C     METHOD
C
  110 CALL SECOND (TIM15)
```

```
C
C
C     FLAGS FOR PRINTING, SAVING NODAL AND ELEMENT RESPONSES
C        KPRI MASTER CONTROL = .EQ.0  STRESS CALCULATIONS
C                                  FOR PRINTING OR SAVING PURPOSES ONLY
C        IPRI.EQ.0    FOR PRINT-OUT OF DISP.VEL.ACC AND STRESSES
C        KPLOTN.EQ.0 FOR SAVING NODAL DISP, VEL, ACC VECTORS
C        KPLOTE.EQ.0 FOR SAVING ELEMENT STRESSES
C
      CALL BLKCNT(KSTEP,NPRIB,IPRI,IPRIB,NSTE,3)
      CALL BLKCNT(KSTEP,NODSVB,KPLOTN,INODB,NSTE,4)
      CALL BLKCNT(KSTEP,LEMSVB,KPLOTE,IELMB,NSTE,5)
      KPRI=IPRI
      IF (KPRI.NE.0) KPRI=KPLOTE
      CALL NEWDAV (A(N4),A(N3),A(N1),A(N2),A(N3),A(N7),A(N8),1)
C
      CALL SECOND (TIM16)
      TSUM5=TSUM5 + (TIM16 - TIM15)
      IF ( IPRI.EQ.0) WRITE (6,2020) KSTEP,TIME
      IF ( IPRI .NE. 0) GO TO 171
      IF (IEQUIT.EQ.0) WRITE (6,2060) ITE
      IF ( IEQUIT.GT.0) WRITE (6,2050)
      IF ( IREF.EQ.0) WRITE(6,2070)
      IF ( IREF.NE.0) WRITE(6,2080)
  171 IF (KPRI .NE. 0) GO TO 170
C
C     PRINT DISPLACEMENTS,VELOCITIES AND ACCELERATIONS
C
      IF (IPRI.EQ.0 .AND. IPC.NE.0)
     1  CALL WRITE (A(N1),A(N2),A(N7),A(N8),A(N5),NEQ,NDOF,2)
C
      CALL SECOND (TIM17)
      TSUM5=TSUM5 + (TIM17 - TIM16)
C
C
C     C A L C U L A T I O N   O F   S T R E S S E S
C
C
C
      CALL SECOND (TIM18)
      TSUM6=TSUM6 + (TIM18 - TIM17)
      KPRI=1
C
C     UPDATE DISPLACEMENT VECTORS, IF CENTRAL DIFFERENCE METHOD IS USED
C
  170 IF (IOPE.EQ.3)
     1    CALL NEWDAV (A(N4),A(N3),A(N1),A(N2),A(N3),A(N7),A(N8),2)
C
C     SHIFT TEMPERATURE ARRAY (IF APPLICABLE)
C
      IF (ITEMPR.LT.2) GO TO 180
      DO 174 I=1,NUMP1
  174 A(N6A+I-1)=A(N6B+I-1)
C
C
C     P R E P A R E   T A P E S   F O R   P O S S I B L E
C                     F U T U R E   R E S T A R T   J O B
C
```

```
C
C       FLAG FOR SAVING RESTART INFORMATION
C          IRR.EQ.0 SAVE INFORMATION
C          IRR.GT.0 NO SAVE
C
  180   KRINT=KRINT + 1
        IRR= IRINT - KRINT
        IF (KSTEP.EQ.NSTE) IRR=0
        IF (IRR.GT.0) GO TO 120
        KRINT=0
C
        CALL RSTART (A(N1),A(N2),A(N7),A(N8),A(N10),NEQ,IOPE,1)
C
  120   IF (KSTEP.LT.NSTE) GO TO 100
C
C
C       P R I N T   T I M E   L O G
C
  190   CALL SECOND (TIM19)
        WRITE(6,2090) HED
        TIM10=TIM2 - TIM1
        TIM11=TIM4 - TIM3
        TIM12=TIM6 - TIM5
        TIM13=TIM8 - TIM7
        TIM14=TIM19 - TIM8
        TIM15=TIM19 - TIM1
        WRITE (6,2100) TIM10,TIM11,TIM12,TIM13,KSTEP,TSUM1,TSUM2,TSUM3,
     +               TSUM4,TSUM5,TSUM6,TIM14,TIM15
C
C         * * * * * * *        R A N D O M   A C C E S S        * * * *
C
C *CDC*         IF (KLIN.NE.0) CALL CLOSMS(2)
C *CDC*         IF (IOPE.NE.3) CALL CLOSMS (10)
C
C         * * * * * * *        R A N D O M   A C C E S S        * * * *
C
        GO TO 200
C
C
 2020 FORMAT (1H1,46H P R I N T   O U T   F O R   T I M E   S T E P ,I5,
     1 40X,12H ( AT TIME  ,E10.4,2H ) )
 2030 FORMAT(////41H  D A T A   C H E C K   C O M P L E T E D)
 2040 FORMAT (//// 64H STOP BECAUSE OUT-OF-BALANCE LOADS LARGER THAN INC
     1REMENTAL LOADS.)
 2050 FORMAT (/2X,44HNO EQUILIBRIUM ITERATION IN THIS TIME STEP         )
 2060 FORMAT (/1X,I5,79H EQUILIBRIUM ITERATIONS PERFORMED IN THIS TIME
     1STEP TO REESTABLISH EQUILIBRIUM   )
 2070 FORMAT (2X,48HSTIFFNESS REFORMED FOR THIS TIME STEP              )
 2080 FORMAT (2X,42HSTIFFNESS NOT REFORMED FOR THIS TIME STEP           )
 2090 FORMAT (1H1,44H S O L U T I O N   T I M E   L O G (IN SEC) //12X,
     1              11HFOR PROBLEM//1X,12A6////)
 2100 FORMAT (49H INPUT PHASE  . . . . . . . . . . . . . . . . . . .F9.2//
     1              49H ASSEMBLAGE OF LINEAR STIFFNESS,EFFECTIVE STIFF-     /
     2              49H NESS,MASS MATRICES AND LOAD VECTORS. . . . . . . . F9.2//
     3              49H FREQUENCY ANALYSIS . . . . . . . . . . . . . . . .F9.2//
     4              49H TRIANGULARIZATION OF LINEAR (EFFECTIVE)              /
     +              49H                   STIFFNESS MATRIX  . . . . .F9.2//
     5              24H STEP-BY-STEP SOLUTION (,I5,12H TIME STEPS)          //
```

```
     6          43H    CALCULATION OF EFFECTIVE LOAD VECTORS. .        ,F9.2/
     7          43H    UPDATING EFFECTIVE STIFFNESS MATRICES                /
     8          43H      AND LOAD VECTORS FOR NONLINEARITIES . ........,F9.2/
     9          43H    SOLUTION OF EQUATIONS . . . . . . . . . ........,F9.2/
     A          43H    EQUILIBRIUM ITERATIONS . . . . . . . . . .......,F9.2/
     B          43H    CALCULATION AND PRINTING OF DISPLACE-                /
     C          43H      MENTS, VELOCITIES, AND ACCELERATIONS.          ,F9.2/
     D          43H    CALCULATION AND PRINTING OF STRESSES . ....      F9.2//
     E          30X,19H STEP-BY-STEP TOTAL ,F9.2/////
     F          49H T O T A L   S O L U T I O N   T I M E (SEC) . . ,F9.2)
 2200 FORMAT (//30H STORAGE CHECK FOR LOAD INPUT )
 2202 FORMAT(//53H STORAGE CHECK FOR MATRIX ASSEMBLAGE,TIME INTEGRATION)
 2203 FORMAT (//60H  ** STOP ** NO STORAGE AVAILABLE TO STORE STIFFNESS
     1MATRIX. ,/68H INCREASE MTOT AND/OR BREAK ELEMENTS INPUT INTO MORE
     2ELEMENT GROUPS.  )
 2204 FORMAT (// 22H STOP  ERROR IN INPUT    //
     1           33H MORE THAN 1000 SOLUTION BLOCKS REQD     )
 2210 FORMAT (1H1,                                                 ///5X,
     155HTOTAL SYSTEM DATA
     255HNUMBER OF EQUATIONS . . . . . . . . . . . . . . .(NEQ)   =,I8//5X,
     355HNUMBER OF MATRIX ELEMENTS . . . . . . . . . . . .(NWK)   =,I8//5X,
     455HMAXIMUM HALF BANDWIDTH . . . . . . . . . . . . . (MA )   =,I8//5X,
     555HMEAN HALF BANDWIDTH . . . . . . . . . . . . . . .(MAM)   =,I8//5X,
     655HMAXIMUM BLOCK LENGTH . . . . . . . . . . . . . (ISTOH)   =,I8//5X,
     755HNUMBER OF BLOCKS . . . . . . . . . . . . . . .(NBLOCK)   =,I8//5X,
     855HMAXIMUM TOTAL STORAGE AVAILABLE. . . . . . . .( MTOT )   =,I8//)
 2220 FORMAT(4X,51H NUMBER OF COLUMNS PER BLOCK AND 1ST COUPLING BLOCK)
 2230 FORMAT (6X,16H NUMBER OF BLOCK,12X,(21I4,/34X))
 2240 FORMAT (6X,28H NUMBER OF COLUMNS PER BLOCK,(21I4,/,34X))
 2250 FORMAT (6X,21H FIRST COUPLING BLOCK,7X,(21I4,/,34X))
 2260 FORMAT (//50H0**STORAGE CHECK FOR ASSEMBLAGE OF LINEAR MATRICES )
 2270 FORMAT (//50H0**STORAGE CHECK FOR LOAD VECTORS INPUT PHASE      )
 2280 FORMAT (//50H0**STORAGE CHECK FOR TIME INTEGRATION PHASE        )
 2290 FORMAT (//15H  STEP NUMBER =,I5,5X,12H ( AT TIME  ,E10.4,2H )   )
 2300 FORMAT (//,59H TIME AT ENTERING EQUATION SOLVER FOR FIRST FACTORIS
     1ATION =,F10.5)
 2310 FORMAT (//,67H TIME AT THE END OF SOLUTION OF EQUATIONS FOR THE FI
     1RST TIME STEP =,F10.5)
C     DEBUG TRACE,SUBTRACE
C     AT 100
C     TRACE ON
C     AT 190
C     TRACE OFF
      END
```

```
FORTRAN IV G1  RELEASE 2.0                    MAIN                DATE = 81223

    *OPTIONS IN EFFECT*  NOTERM,ID,EBCDIC,SOURCE,NOLIST,NODECK,LOAD,NOMAP
    *OPTIONS IN EFFECT*  NAME = MAIN    , LINECNT =          60
    *STATISTICS*     SOURCE STATEMENTS =        343,PROGRAM SIZE =    14568
    *STATISTICS*  NO DIAGNOSTICS GENERATED
```

MAIN DATE = 81229 09/18/44

```
C             DATA SET ADINI      AT LEVEL TMP AS OF 08/17/81
C *CDC* *DECK OVL1O
C *CDC*         OVERLAY (ADINA,1,0)
C *CDC* *DECK ADINI
C *UNI* )FOR,IS  N.ADINI, R.ADINI
C *CDC*         PROGRAM ADINI
      SUBROUTINE ADINI
C
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /SOL/ NUMNP,NEQ,NWK,NWM,NWC,NUMEST,MIDEST,MAXEST,NSTE,MA
      COMMON /DIM/ N0,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /EL/ IND,ICOUNT,NPAR(20),NUMEG,NEGL,NEGNL,IMASS,IDAMP,ISTAT
     1          ,NDOF,KLIN,IEIG,IMASSN,IDAMPN
      COMMON /CONST/ DT,DTA,A0,A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11
     1          ,A12,A13,A14,A15,A16,A17,A18,A19,A20,IOPE
      COMMON /LOA/ TEND,NTFN,NPTM,NLOAD,IDGRAV,NPR2,NPR3,NODE3
      COMMON /JUNK/ HED(12),MTOT,LPROG
      COMMON /VAR/ NG,MODEX,IUPDT,KSTEP,ITEMAX,IEQREF,ITE,KPRI,
     1          IREF,IEQUIT,IPRI,KPLOTN,KPLOTE
      COMMON /NORMS/ RNORM, RENORM, RTOL,DNORMM
      COMMON /PRCON/ IDATWR,IPRIC,NPB,IDC,IVC,IAC,IPC,IPNODE(3,15)
      COMMON /ADINAI/ OPVAR(7),TSTART,IRINT,ISTOTE
      COMMON /TEMPRT/ TEMP1,TEMP2,ITEMPR,ITP96,N6A,N6B
      COMMON /MDFROM/ IDOF(6)
      COMMON /DPR/ ITWO
      COMMON /BLOCKS/ NSREFB,NEQITB,NPRIB,NODSVB,LEMSVB,ISREFB(3,10),
     1          IEQITB(3,10),IPRIB(3,10),INODB(3,10),IELMB(3,10)
      COMMON /PORT/ INPORT,JNPORT,NPUTSV,LUNODE,LU1,LU2,LU3,JDC,JVC,JAC
      COMMON /RANDI/ NOA,NID,IELCPL
      COMMON /BEAR/ IBEAR,MTOTB
      COMMON A(1)
      REAL A
      DIMENSION BLKNAM(5)
      DATA BLKNAM /8HSTIFNESS, 8HITERATON, 8HPRINTOUT,
     1          8HNODESAVE, 8HELMTSAVE /
      DATA RECLB1 /8HMASTERCP/, FINAL /6HSTOP  /
C
C     IND.EQ... WHEN LOAD VECTORS ARE TO BE ASSEMBLED
C
      IF (IND.EQ.2) GO TO 30
C
C
C     R E A D   C O N T R O L   I N F O R M A T I O N
C
C
      READ (5,1000) HED
      IF (HED(1).EQ.FINAL) STOP
      READ (5,1003) NUMNP,(IDOF(I),I=1,6),NEGL,NEGNL,MODEX,NSTE,
     1 DT,TSTART,IDATWR,IRINI,ITP96,INPORT,JNPORT,ISTOTE
      IF (NUMNP.EQ.0) STOP
      IF (IDATWR.EQ.0 .OR. IDATWR.EQ.2) CALL INLIST (2)
      READ (5,1001) IMASS,IDAMP,IMASSN,IDAMPN
C.......................................................................
C.                                                                    .
C.    R E A D   C A L C U L A T I O N   M O D E                       .
C.                                                                    .
C.           IBEAR = 1    BEARING CALCULATIONS REQUIRED               .
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
C..
C.              IBEAR = 0   NO BEARING CALCULATIONS REQUIRED                 .
C.                                                                            .
C..............................................................................
      READ (5,1002) IFIG,IBEAR
      WRITE (6,4000) IBEAR
 4000 FORMAT (////5X,'IBEAR(ADINI)=',I5)
      READ (5,1020) IOPE,OPVAR
      READ(5,1005) NSREFB,NEQITB,ITEMAX,RTOL
      READ(5,1010) NPRIB,NPB,IDC,IVC,IAC
      READ (5,1010) NPUTSV,NODSVB,LEMSVB,LUNODE,LU1,LU2,LU3,JDC,JVC,JAC
C
      IF (NSREFB.EQ.0) GO TO 250
      READ (5,1100)((ISREFB(I,J),I=1,3),J=1,NSREFB)
      IF (NSTE.GT.0 .AND. ISREFB(1,1) .EQ. 0) ISREFB(1,1) = 1
      IF (ISREFB(2,1) .EQ. 0) ISREFB(2,1) = NSTE
      IF (ISREFB(3,1) .EQ. 0) ISREFB(3,1) = 1
      INDEX=1
      IF (NSREFB.LE.1) GO TO 240
      DO 230 I=2,NSREFB
      J=I - 1
      IF (ISREFB(1,J).GT.ISREFB(2,J)) GO TO 235
      IF (ISREFB(1,I).GE.ISREFB(2,J)) GO TO 230
      WRITE (6,3000) BLKNAM(INDEX),I,J
      STOP
  230 CONTINUE
  240 J=NSREFB
      IF (ISREFB(1,J).LE.ISREFB(2,J)) GO TO 245
  235 WRITE (6,3004) BLKNAM(INDEX),J,J
      STOP
  245 IF (ISREFB(2,NSREFB).GE.NSTE) GO TO 250
      WRITE (6,3001) BLKNAM(INDEX),ISREFB(2,NSREFB),NSTE
      STOP
C
  250 IF (NEQITB.EQ.0) GO TO 350
      READ (5,1100)((IEQITB(I,J),I=1,3),J=1,NEQITB)
      IF (NSTE.GT.0 .AND. IEQITB(1,1) .EQ. 0) IEQITB(1,1) = 1
      IF (IEQITB(2,1) .EQ. 0) IEQITB(2,1) = NSTE
      IF (IEQITB(3,1) .EQ. 0) IEQITB(3,1) = 1
      INDEX=2
      IF (NEQITB.LE.1) GO TO 340
      DO 330 I=2,NEQITB
      J=I - 1
      IF (IEQITB(1,J).GT.IEQITB(2,J)) GO TO 335
      IF (IEQITB(1,I).GE.IEQITB(2,J)) GO TO 330
      WRITE (6,3000) BLKNAM(INDEX),I,J
      STOP
  330 CONTINUE
  340 J=NEQITB
      IF (IEQITB(1,J).LE.IEQITB(2,J)) GO TO 345
  335 WRITE (6,3004) BLKNAM(INDEX),J,J
      STOP
  345 IF (IEQITB(2,NEQITB).GE.NSTE) GO TO 350
      WRITE (6,3001) BLKNAM(INDEX),IEQITB(2,NEQITB),NSTE
      STOP
C
  350 IF (NPRIB .EQ.0) GO TO 450
      READ (5,1100)((IPRIB(I,J),I=1,3),J=1,NPRIB)
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
      IF (NSTE.GT.0 .AND. IPRIB(1,1).EQ.0) IPRIB(1,1)=1
      IF ( IPRIB(2,1) .EQ. 0)  IPRIB(2,1) = NSTE
      IF ( IPRIB(3,1) .EQ. 0)  IPRIB(3,1) = 1
      INDEX=3
      IF (NPRIB.LE.1) GO TO 440
      DO 430 I=2,NPRIB
      J=I - 1
      IF (IPRIB(1,J).GT.IPRIB(2,J)) GO TO 435
      IF (IPRIB(1,I).GE.IPRIB(2,J)) GO TO 430
      WRITE (6,3000) BLKNAM(INDEX),I,J
      STOP
  430 CONTINUE
  440 J=NPRIB
      IF (IPRIB(1,J).LE.IPRIB(2,J)) GO TO 445
  435 WRITE (6,3004) BLKNAM(INDEX),J,J
      STOP
  445 IF (IPRIB(2,NPRIB).GE.NSTE) GO TO 450
      WRITE (6,3001) BLKNAM(INDEX),IPRIB(2,NPRIB),NSTE
      STOP
C
  450 IF (NPB    .EQ.0) GO TO 550
      READ (5,1100) ((IPNODE(I,J),I=1,3),J=1,NPB)
      DO 500 I=1,NPB
  500 IF (IPNODE(3,I).EQ.0) IPNODE(3,I)=1
  550 IF (JNPORT.EQ.0 .OR. NODSVB.EQ.0) GO TO 650
      READ (5,1100)((INODB(I,J),I=1,3),J=1,NODSVB)
      IF (NSTE.GT.0 .AND.  INODB(1,1) .EQ. 0)  INODB(1,1) = 1
      IF ( INODB(2,1) .EQ. 0)  INODB(2,1) = NSTE
      IF ( INODB(3,1) .EQ. 0)  INODB(3,1) = 1
      INDEX=4
      IF (NODSVB.LE.1) GO TO 640
      DO 630 I=2,NODSVB
      J=I - 1
      IF (INODB(1,J).GT.INODB(2,J)) GO TO 635
      IF (INODB(1,I).GE.INODB(2,J)) GO TO 630
      WRITE (6,3000) BLKNAM(INDEX),I,J
      STOP
  630 CONTINUE
  640 J=NODSVB
      IF (INODB(1,J).LE.INODB(2,J)) GO TO 645
  635 WRITE (6,3004) BLKNAM(INDEX),J,J
      STOP
  645 IF (INODB(2,NODSVB).GE.NSTE) GO TO 650
      WRITE (6,3001) BLKNAM(INDEX),INODB(2,NODSVB),NSTE
      STOP
C
  650 IF (JNPORT.EQ.0 .OR. LEMSVB.EQ.0) GO TO 750
      READ (5,1100)((IELMB(I,J),I=1,3),J=1,LEMSVB)
      IF (NSTE.GT.0 .AND. IELMB(1,1) .EQ. 0)  IELMB(1,1) = 1
      IF ( IELMB(2,1) .EQ. 0)  IELMB(2,1) = NSTE
      IF ( IELMB(3,1) .EQ. 0)  IELMB(3,1) = 1
      INDEX=5
      IF (LEMSVB.LE.1) GO TO 740
      DO 730 I=2,LEMSVB
      J=I - 1
      IF (IELMB(1,J).GT.IELMB(2,J)) GO TO 735
      IF (IELMB(1,I).GE.IELMB(2,J)) GO TO 730
      WRITE (6,3000) BLKNAM(INDEX),I,J
```

```
      STOP
  730 CONTINUE
  740 J=LEMSVB
      IF (IELMB(1,J).LE.IELMB(2,J)) GO TO 745
  735 WRITE (6,3004) BLKNAM(INDEX),J,J
      STOP
  745 IF (IELMB(2,LEMSVB).GE.NSTE) GO TO 750
      WRITE (6,3001) BLKNAM(INDEX),IELMB(2,LEMSVB),NSTE
      STOP
  750 CONTINUE
C
C     VERIFY AND INITIALIZE SOLUTION VARIABLES
C
      NUMEG=NEGL + NEGNL
      NDOF=6
      DO 1 I=1,6
    1 NDOF=NDOF - IDOF(I)
      ISTAT=1
      IF (IMASS.EQ.0) ISTAT=0
      IF (IOPE.NE.3 .OR. IMASS.EQ.1) GO TO 20
      WRITE (6,3012)
      STOP
   20 IF (IDAMP.EQ.1 .AND. IMASS.EQ.0) GO TO 5
      IF (IDAMPN.NE.0 .AND. IMASS.EQ.0) GO TO 5
      IF (IMASSN.NE.0 .AND. IMASS.EQ.0) GO TO 5
      GO TO 3
    5 WRITE(6,3002)
      STOP
    3 TH=1.0
      DTA=DT
      KLIN=1
      IF (NEGNL.EQ.0) KLIN=0
      IF (IOPE.NE.3 .OR. IEIG.EQ.0) GO TO 2
      WRITE (6,3010)
      STOP
    2 IF (IEIG.LE.1) GO TO 4
      WRITE(6,3003)
      STOP
    4 IF (ITEMAX.EQ.0) ITEMAX=15
      IF (RTOL.EQ.0) RTOL=1.0E-03
      IF (IOPE.EQ.3 .OR. KLIN.EQ.0) NSREFB=0
      IF (IOPE.EQ.3 .OR. KLIN.EQ.0) NEQITB=0
      IPC= IDC+IVC+IAC
      IF (NPB.NE.0) GO TO 6
      IPC=1
      NPB=1
      IDC=1
      IVC=1
      IAC=1
      IPNODE(1,1)=1
      IPNODE(2,1)=NUMNP
      IPNODE(3,1)=1
    6 CONTINUE
      IF (IOPE.EQ.0) IOPE=2
      IF (IRINT.EQ.0) IRINT=9999
C
      A0=0.
      A1=0.
```

```
      IF(MODEX.NE.2 .AND. JNPORT.GT.0) NPUTSV=1
      IF (LUNODE .EQ. 0) LUNODE=60
      IF (LU1    .EQ. 0) LU1    =61
      IF (LU2    .EQ. 0) LU2    =62
      IF (LU3    .EQ. 0) LU3    =63
C
C***  DATA PORTHOLE (START)
C
      IF (JNPORT.EQ.0) GO TO 790
      RECLAB=RECLB1
      WRITE (LUNODE) RECLAB,(HED(I),I=1,12),NUMNP,(IDOF(I),I=1,6),
     1               NEGL,NEGNL,MODEX,NSTE,DT,TSTART,IDATWR,IRINT,
     2               ITP96,INPORT,JNPORT,IMASS,IDAMP,IMASSN,IDAMPN,
     3               IEIG,NSREFB,NEQITB,RTOL,ITEMAX,IOPE,OPVAR(1),
     4               OPVAR(2),NPRIB,NODSVB,LEMSVB,LUNODE,LU1,LU2,LU3,
     5               NPB,IDC,IVC,IAC,NPUTSV,JDC,JVC,JAC,
     6               ((IPNODE(I,J),I=1,3),J=1,NPB)
      RECLAB=BLKNAM(1)
      IF (NSREFB.NE.0) WRITE(LUNODE) RECLAB,((ISREFB(I,J),I=1,3),
     1   J=1,NSREFB)
      RECLAB=BLKNAM(2)
      IF (NEQITB.NE.0) WRITE(LUNODE) RECLAB,((IEQITB(I,J),I=1,3),
     1   J=1,NEQITB)
      RECLAB=BLKNAM(3)
      IF (NPRIB.NE.0) WRITE(LUNODE) RECLAB,((IPRIB(I,J),I=1,3),
     1   J=1,NPRIB)
      RECLAB=BLKNAM(4)
      IF (NODSVB.NE.0) WRITE(LUNODE) RECLAB,((INODB(I,J),I=1,3),
     1   J=1,NODSVB)
      RECLAB=BLKNAM(5)
      IF (LEMSVB.NE.0) WRITE(LUNODE) RECLAB,((IELMB(I,J),I=1,3),
     1   J=1,LEMSVB)
C
C***  DATA PORTHOLE (END)
C
  790 WRITE (6,2040) HED
      IF (ISTAT.EQ.0) GO TO 10
C
C     SET TIME INTEGRATION COEFFICIENTS IN CASE OF DYNAMIC PROBLEM
C
      CALL OPCOEF(OPVAR)
C
   10 IF (IOPE.EQ.1) TH=OPVAR(1)
C
      IF (IDATWR.GT.1) GO TO 890
      WRITE(6,2045)
      NCARD=1
      WRITE(6,2055) NCARD
      WRITE(6,2000) NUMNP,(IDOF(I),I=1,6),NEGL,NEGNL,MODEX
      WRITE (6,2002) NSTE,DT,TSTART,IDATWR
      WRITE (6,2023) IRINT,ITP96,INPORT,JNPORT
      NCARD=2
      WRITE(6,2055) NCARD
      WRITE(6,2008) IMASS,IDAMP,IMASSN,IDAMPN
      NCARD=3
      WRITE(6,2055) NCARD
      WRITE(6,2001) IEIG
      NCARD = 4
```

```
      WRITE (6,2055) NCARD
      WRITE (6,2005) IOPE
      IF (IOPE.EQ.1) WRITE(6,2006) OPVAR(1)
      IF (IOPE.EQ.2) WRITE(6,2007) OPVAR(1),OPVAR(2)
      NCARD=5
      WRITE(6,2055) NCARD
      WRITE (6,2003) NSREFB,NEQITB,ITEMAX,RTOL
C
C     PRINT DISPL/VEL/ACC OUTPUT INFORMATION
C
      NCARD=6
      WRITE(6,2055) NCARD
      WRITE (6,2012) NPRIB,NPB,IDC,IVC,IAC
      NCARD=7
      WRITE(6,2055) NCARD
      WRITE (6,2100) NPUTSV,NODSVB,LEMSVB
      WRITE (6,2103) LUNODE,LU1,LU2,LU3,JDC,JVC,JAC
      IF (NSREFB.NE.0 .OR. NEQITB.NE.0) GO TO 810
      IF ( NPRIB.NE.0 .OR. JNPORT.NE.0) GO TO 810
      IF (NPB.NE.1 .OR. (IPNODE(1,1).NE.1 .OR. IPNODE(2,1).NE.NUMNP))
     1 GO TO 810
      GO TO 890
  810 WRITE (6,2110)
  820 IF (NSREFB.EQ.0) GO TO 830
      WRITE (6,2120)
      WRITE (6,2130) (J,(J,ISREFB(I,J),I=1,3),J=1,NSREFB)
  830 IF (NEQITB.EQ.0) GO TO 840
      WRITE (6,2140)
      WRITE (6,2150) (J,(J,IEQITB(I,J),I=1,3),J=1,NEQITB)
  840 IF (NPRIB.EQ.0) GO TO 850
      WRITE (6,2160)
      WRITE (6,2170) (J,(J, IPRIB(I,J),I=1,3),J=1,NPRIB)
  850 IF (IPNODE(1,1).EQ.1 .AND. IPNODE(2,1).EQ.NUMNP) GO TO 860
      WRITE (6,2180)
      WRITE (6,2190) (J,(J,IPNODE(I,J),I=1,3),J=1,NPB)
  860 IF (NODSVB.EQ.0) GO TO 870
      WRITE (6,2200)
      WRITE (6,2210) (J,(J, LNODB(I,J),I=1,3),J=1,NODSVB)
  870 IF (LEMSVB.EQ.0) GO TO 880
      WRITE (6,2220)
      WRITE (6,2230) (J,(J, IELMB(I,J),I=1,3),J=1,LEMSVB)
  880 WRITE (6,2004) ISTAT,KLIN
C
  890 CONTINUE
      RTOL=RTOL * RTOL
      DNORMM=0.00001
C
C
C     R E A D   N O D A L   P O I N T   D A T A
C
C
      NOA=NO + NEGNL
      N1=NOA + NEGNL + 1
      N2=N1 + NDOF*NUMNP
      N3=N2 + NUMNP*ITWO
      N4=N3 + NUMNP*ITWO
      N5=N4 + NUMNP*ITWO
      CALL SIZE(N5)
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
C
      CALL INPUT (A(N1),A(N2),A(N3),A(N4),NUMNP,NDOF,NEQ,MODEX)
C
      N3=N2 + NEQ*ITWO
      N4=N3 + NEQ*ITWO
      N5=N4 + NEQ*ITWO
      IF (ISTAT.EQ.0) N5=N3
      N6=N5 + NEQ*ITWO
      CALL SIZE (N6)
C
C     R E A D   L O A D   V E C T O R   C O N T R O L
C                                       I N F O R M A T I O N
C
      READ (5,1010) NLOAD,NTFN,NPTM,IDGRAV,NPR2,NPR3,IPRTYP
      IF (IDATWR.LE.1)
     1 WRITE (6,2010) NLOAD,NTFN,NPTM,IDGRAV,NPR2,NPR3,IPRTYP
      IF (IDGRAV.EQ.0 .OR. ISTAT.EQ.0) GO TO 25
      IF (IMASS.EQ.1) GO TO 25
      WRITE (6,3020)
      STOP
C
   25 NODE3=4
      IF (IPRTYP.EQ.1) NODE3=8
      IF (IPRTYP.EQ.2) NODE3=12
C
C
C     E S T A B L I S H   C O N C E N T R A T E D   N O D A L
C         M A S S   A N D   D A M P I N G   V E C T O R S
C
C
      CALL NODMAS  (A(N1),A(N5),NDOF)
C
C
C     R E A D . I N I T I A L   C O N D I T I O N S
C
C
      CALL INITAL (A(N2),A(N5),A(N3),A(N4),A(N1),NDOF)
C
C     REINSTATE NODAL CORDINATES INTO HIGH SPEED STORAGE FROM TAPE9
C
      REWIND 9
      N3=N2 + NUMNP*ITWO
      N4=N3 + NUMNP*ITWO
      N5=N4 + NUMNP*ITWO
      NN=N3 - 1
      READ (9) (A(I),I=N2,NN)
      NN=N4 - 1
      READ (9) (A(I),I=N3,NN)
      NN=N5 - 1
      READ (9) (A(I),I=N4,NN)
C
      GO TO 599
C
C
C
C     C A L C U L A T E   A N D   S T O R E   L O A D   V E C T O R S
C
C
```

```
C *CDC*      30 CALL OVERLAY (5HADINA,1,1,5HRECALL)
   30 CALL LOAD
C
  599 CONTINUE
      RETURN
 1000 FORMAT (12A6)
 1001 FORMAT (4I5)
 1002 FORMAT (2I5)
 1003 FORMAT (I5,6I1,I4,3I5,2F10.0,6I5)
 1005 FORMAT (3I5,F10.0)
 1010 FORMAT (16I5)
 1020 FORMAT (I10,7F10.0)
 1100 FORMAT (15I5,5X )
C
 2000 FORMAT(/,5X,
     255HNUMBER OF NODAL POINTS . . . . . . . . . . . . (NUMNP)  =,I5//5X,
     355HMASTER X-TRANSLATION CODE . . . . . . . . . . (IDOF(1)) =,I5//5X,
     355HMASTER Y-TRANSLATION CODE . . . . . . . . . . (IDOF(2)) =,I5//5X,
     355HMASTER Z-TRANSLATION CODE . . . . . . . . . . (IDOF(3)) =,I5//5X,
     355HMASTER X-ROTATION    CODE . . . . . . . . . . (IDOF(4)) =,I5//5X,
     355HMASTER Y-ROTATION    CODE . . . . . . . . . . (IDOF(5)) =,I5//5X,
     355HMASTER Z-ROTATION    CODE . . . . . . . . . . (IDOF(6)) =,I5//5X,
     455HNUMBER OF LINEAR ELEMENT GROUPS . . . . . . . (NEGL)    =,I5//5X,
     555HNUMBER OF NONLINEAR ELEMENT GROUPS . . . . . (NEGNL)   =,I5//5X,
     D55HSOLUTION MODE . . . . . . . . . . . . . . . . (MODEX)   =,I5 /5X,
     E55H    EQ.0, DATA CHECK                                          /5X,
     F55H    EQ.1, EXECUTION                                           /5X,
     G55H    EQ.2, RESTART                                             /5X,
     )
 2001 FORMAT (/5X,
     H55HFREQUENCIES SOLUTION CODE . . . . . . . . . . (IEIG)    =,I5 /5X,
     I55H    EQ.0, NO FREQUENCIES SOLUTION                             /5X,
     J55H    EQ.1, FREQUENCIES AND MODE SHAPES                         /5X,
     K55H          ARE DETERMINED
     )
 2002 FORMAT (/4X,
     A56H NUMBER OF TIME STEPS . . . . . . . . . . . . (NSTE)      =I5//4X,
     B56H TIME STEP INCREMENT . . . . . . . . . . . . (DT)          =E11.4//
     C 4X,
     D56H TIME AT SOLUTION START . . . . . . . . . . . (TSTART)   =E11.4//
     E 4X,
     F 55H FLAG FOR WRITING INPUT DATA IN CARD IMAGE AND/OR         /5X,
     G 55H   GENERATED FORM . . . . . . . . . . . . . . (IDATWR)  =I5 /4X,
     H 55H      EQ.0, BOTH CARD IMAGE LISTING AND DETAILED           /4X,
     I 55H            OUTPUT OF INPUT DATA                           /4X,
     J 55H      EQ.1, ONLY DETAILED OUTPUT OF INPUT DATA              /4X,
     K 55H      EQ.2, ONLY A CARD IMAGE LISTING OF INPUT DATA         /4X,
     L 55H      GT.2, NO DETAILED OUTPUT NOR CARD IMAGE               /4X,
     M 55H            LISTING OF INPUT DATA                           /4X,
     )
 2020 FORMAT (/5X,
     T55HRESTART SAVE INTERVAL . . . . . . . . . . . (IRINT)      =,I5//5X,
     U55HTEMPERATURE TAPE FLAG . . . . . . . . . . . (ITP96)      =,I5/5X,
     V35H    EQ.0, TEMPERATURE TAPE NOT USED,/5X,
     W30H    EQ.1, TEMPERATURE TAPE USED   ,//5X,
     A 55HPREPROCESSOR INPUT CONTROL PARAMETER . . . . (INPORT)   =I5 /4X,
     B 55H      EQ.0, NO PREPROCESSOR TAPE USED                       /4X,
     C 56H      EQ.1, NODAL AND ELEMENT INFORMATION READ FROM UNIT59  /4X,
     D 56H      EQ.2, ABOVE INFORMATION AND ID ARRAY READ FROM UNIT59 //5X
     R 55HPORTHOLE PARAMETER . . . . . . . . . . . . (JNPORT)    =I5 /4X,
     S 55H      EQ.0, PORTHOLE NOT WRITTEN                            /4X,
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
      T 55H     EQ.1. PORTHOLE WRITTEN                                       //)
 2003 FORMAT (/,4X,
     A 55H NO. OF BLOCKS OF EFFECTIVE STIFFNESS                              /4X,
     B56H  REFORMATION TIME STEPS . . . . . . . . . . .(NSREFB)   =I5 /4X,
     C 55H     EQ.0. NO STIFFNESS REFORMATION                               //4X,
     D 55H NO. OF BLOCKS OF EQUILIBRIUM                                      /4X,
     E56H  ITERATION TIME STEPS . . . . . . . . . . . . .(NEQITB)  =I5 /4X,
     F 55H     EQ.0. NO EQUILIBRIUM ITERATION PERFORMED                     //5X,
     X55HMAXIMUM NUMBER OF EQUILIBRIUM                                       /5X,
     Y55H  ITERATIONS PERMITTED . . . . . . . . . . . . (ITEMAX)  =,I5//5X,
     1 55H CONVERGENCE TOLERANCE . . . . . . . . . . . . (RTOL)    =E11.4 )
 2004 FORMAT (1H0,////,1X,
     A62HA N A L Y S I S  T Y P E   (ESTABLISHED USING IMASS,NEGL,NEGNL),
     G//,5X,
     B55HTIME DEPENDENCY CODE . . . . . . . . . . . . . (ISTAT)    =,I5 /5X,
     C55H    EQ.0. STATIC ANALYSIS                                           /5X,
     D55H    EQ.1. DYNAMIC ANALYSIS                                         //5X,
     E55HNONLINEARITY CODE . . . . . . . . . . . . . . . (KLIN)     =,I5 /5X,
     F55H    EQ.0. LINEAR ANALYSIS                                           /5X,
     G55H    EQ.1. NONLINEAR ANALYSIS                                        )
 2005 FORMAT (/5X,
     155HTIME INTEGRATION CODE . . . . . . . . . . . . . (IOPE)     =,I5 /5X,
     255H    EQ.1. WILSONS THETA METHOD                                     ,/,5X,
     355H    EQ.2. NEWMARKS METHOD                                          ,/,5X,
     435H    EQ.3. CENTRAL DIFFERENCE METHOD                                ,/,5X)
 2006 FORMAT (5X,
     155HINTEGRATION PARAMETER . . . . . . . . . . . . . (THETA)    =,F5.2)
 2007 FORMAT (5X,
     155HINTEGRATION PARAMETERS  . . . . . . . . . . . . (DELTA)    =,F5.2/5X
     255H                                                 (ALPHA)    =,F5.2)
 2012 FORMAT (/4X,
     A 55H NO. OF BLOCKS OF TIME STEPS FOR NODAL AND ELEMENT                 /4X,
     B56H  ASSOCIATED QUANTITIES PRINT-OUT . . . . . . .(NPRIB)    =I5 /4X,
     C 55H     EQ.0. NO PRINT-OUT                                           //5X,
     B55HNUMBER OF BLOCKS OF NODAL PRINTOUT . . . . . . . (NPB) =,I5//5X,
     C55HDISPLACEMENT PRINTOUT CODE . . . . . . . . . . . (IDC) =,I5 /5X,
     D55H    EQ.0. NO PRINTING OF DISPLACEMENTS                              /5X,
     E55H    EQ.1. PRINT DISPLACEMENTS                                      //5X,
     F55HVELOCITY PRINTOUT CODE . . . . . . . . . . . . . (IVC) =,I5 /5X,
     G55H    EQ.0. NO PRINTING OF VELOCITIES                                 /5X,
     H55H    EQ.1. PRINT VELOCITIES                                         //5X,
     I55HACCELERATION PRINTOUT CODE . . . . . . . . . . . (IAC) =,I5 /5X,
     J55H    EQ.0. NO PRINTING OF ACCELERATIONS                              /5X,
     K55H    EQ.1. PRINT ACCELERATIONS                                       )
 2008 FORMAT (/5X,
     155HMASS MATRIX CODE . . . . . . . . . . . . . . . .(IMASS)    =,I5 /5X,
     255H    EQ.0. NO MASS EFFECTS                                           /5X,
     355H    EQ.1. LUMPED MASS                                               /5X,
     455H    EQ.2. CONSISTENT MASS                                          //5X,
     555HDAMPING MATRIX CODE . . . . . . . . . . . . . . (IDAMP)    =,I5 /5X,
     655H    EQ.0. NO DAMPING                                                /5X,
     755H    EQ.1. DAMPING INCLUDED                                         //5X,
     855HNUMBER OF NODAL MASSES . . . . . . . . . . . . (IMASSN)   =,I5//5X,
     955HNUMBER OF NODAL DAMPERS . . . . . . . . . . . . (IDAMPN)   =,I5 )
 2010 FORMAT (1H1,34H L O A D   C O N T R O L   D A T A // 4X,
     1    38H NUMBER OF LOADS                         = ,I5 / 4X,
     2    38H NUMBER OF TIME FUNCTIONS                =,I5 / 4X,
     3    38H MAX NUMBER OF POINTS IN LOAD CURVES = ,I5 / 4X,
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
     4   38H GRAVITY LOADING CODE . . .(IOGRAV)   = ,I5 / 4X,
     5   38H    EQ.0. NO GRAVITY LOADING               / 4X,
     6   38H    EQ.1. LUMPED GRAVITY LOADING        ,/,4X,
     7   38H NUMBER OF  2/D PRESSURE LOAD SETS   =,I5 / 4X,
     8   38H NUMBER OF  3/D PRESSURE LOAD SETS   =,I5/,4X,
     9   31H MAX ORDER OF ANY ELEMENT FOR      / ,4X,
     A   25H  3/D PRESSURE LOAD SETS ,12X,1H=,I5///)
 2040 FORMAT(1H1,12A6,///)
 2045 FORMAT(38H M A S T E R   C O N T R O L   C A R D S )
 2055 FORMAT(///,1X,12HCARD NUMBER ,I1)
 2100 FORMAT (/,4X,
     A 55H PORTHOLE CREATION CONTROL CARD                              ///4X,
     B 55H FLAG FOR SAVING INPUT DATA ON TAPE . . . . . (NPUTSV)   =I5 /4X,
     C 55H    EQ.0. WRITE ONLY MAIN HEADER                             /4X,
     D 55H    EQ.1. WRITE ALL INPUT DATA ON PORTHOLE                  //4X,
     E 55H NO. OF BLOCKS OF TIME STEPS FOR SAVING                      /4X,
     F 55H  NODAL RESPONSES ON TAPE . . . . . . . .   . . (NODSVB)   =I5 /4X,
     G 55H    EQ.0. NO ACTION                                         //4X,
     H 55H NO. OF BLOCKS OF TIME STEPS FOR SAVING                      /4X,
     I 55H  ELEMENT RESPONSES ON TAPE . . . . . . . . . (LEMSVB)   =I5 /4X,
     J 55H    EQ.0. NO ACTION                                           /)
 2105 FORMAT (/,4X,
     K 55H NODE DATA SAVE TAPE NUMBER . . . . . . . . . (LUNODE)   =I5//4X,
     L 55H TRUSS/BEAM/TIE  TAPE NUMBER . . . . . . . . . (LU1)      =I5//4X,
     M 55H 2/D CONTINUUM   TAPE NUMBER . . . . . . . . . (LU2)      =I5//4X,
     N 55H 3/D CONTINUUM   TAPE NUMBER . . . . . . . . . (LU3)      =I5//4X,
     A 55H DISPLACEMENT SAVE CODE . . . . . . . . . . . (JDC)      =I5 /4X,
     B 55H    EQ.0. NO SAVING OF DISPLACEMENTS                         /4X,
     C 55H    EQ.1. SAVE DISPLACEMENTS ON PORTHOLE                    //4X,
     D 55H VELOCITY SAVE CODE . . . . . . . . . . . . . (JVC)      =I5 /4X,
     E 55H    EQ.0. NO SAVING OF VELOCITIES                            /4X,
     F 55H    EQ.1. SAVE VELOCITIES ON PORTHOLE                       //4X,
     G 55H ACCELERATION SAVE CODE . . . . . . . . . . . (JAC)      =I5 /4X,
     H 55H    EQ.0. NO SAVING OF ACCELERATIONS                         /4X,
     I 55H    EQ.1. SAVE ACCELERATIONS ON PORTHOLE                      //)
 2110 FORMAT (1H1,5X,48H        S O L U T I O N   D E T A I L   C A R D S
     1 )
 2120 FORMAT (/,5X,90H  (A)   BLOCK DEFINITION CARDS FOR EFFECTIVE STIFF
     1NESS MATRIX REFORMATION TIME STEPS                         )
 2130 FORMAT (/,4X,
     A   7H BLOCK ,I2                                                 //7X,
     B 46H FIRST STEP OF THIS BLOCK  .  .  .   (ISREFB(1,I2,3H))= I5 /7X,
     C 46H LAST  STEP OF THIS BLOCK  .  .  .   (ISREFB(2,I2,3H))= I5 /7X,
     D 46H INCREMENT IN TIME STEP .  .  .  .   (ISREFB(3,I2,3H))= I5 /)
 2140 FORMAT (/,5X,80H  (B)   BLOCK DEFINITION CARDS FOR EQUILIBRIUM ITE
     1RATION TIME STEPS                  )
 2150 FORMAT (/,4X,
     A   7H BLOCK ,I2                                                 //7X,
     B 46H FIRST STEP OF THIS BLOCK  .  .  .   (IEQITB(1,I2,3H))= I5 /7X,
     C 46H LAST  STEP OF THIS BLOCK  .  .  .   (IEQITB(2,I2,3H))= I5 /7X,
     D 46H INCREMENT IN TIME STEP .  .  .  .   (IEQITB(3,I2,3H))= I5 /)
 2160 FORMAT (/,5X,80H  (C)   BLOCK DEFINITION CARDS FOR PRINT-OUT TIME
     1STEPS                              )
 2170 FORMAT (/,4X,
     A   7H BLOCK ,I2                                                 //7X,
     B 46H FIRST STEP OF THIS BLOCK  .  .  .  ( IPRIB(1,I2,3H))= I5 /7X,
     C 46H LAST  STEP OF THIS BLOCK  .  .  .  ( IPRIB(2,I2,3H))= I5 /7X,
     D 46H INCREMENT IN TIME STEP .  .  .  .  ( IPRIB(3,I2,3H))= I5 /)
```

```
2180 FORMAT (/,5X,80H (D) BLOCK DEFINITION CARDS FOR NODAL PARAMETER
    1 PRINT-OUT )
2190 FORMAT (/,4X,                                                  //7X,
    A  7H BLOCK ,I2
    B 46H FIRST NODE OF THIS BLOCK . . . (IPNODE(1,I2,3H))= I5 /7X,
    C 46H LAST  NODE OF THIS BLOCK . . . (IPNODE(2,I2,3H))= I5 /7X,
    D 46H INCREMENT IN NODE NUMBER . . . (IPNODE(3,I2,3H))= I5 /)
2200 FORMAT (/,5X,80H (E) BLOCK DEFINITION CARDS OF TIME STEPS FOR S
    1AVING NODAL RESPONSE                    )
2210 FORMAT (/,4X,                                                  //7X,
    A  7H BLOCK ,I2
    B 46H FIRST STEP OF THIS BLOCK . . . ( INODB(1,I2,3H))= I5 /7X,
    C 46H LAST  STEP OF THIS BLOCK . . . ( INODB(2,I2,3H))= I5 /7X,
    D 46H INCREMENT IN TIME STEP . . . . ( INODB(3,I2,3H))= I5 /)
2220 FORMAT (/,5X,80H (F) BLOCK DEFINITION CARDS OF TIME STEPS FOR S
    1AVING ELEMENT RESPONSES                 )
2230 FORMAT (/,4X,                                                  //7X,
    A  7H BLOCK ,I2
    B 46H FIRST STEP OF THIS BLOCK . . . ( IELMB(1,I2,3H))= I5 /7X,
    C 46H LAST  STEP OF THIS BLOCK . . . ( IELMB(2,I2,3H))= I5 /7X,
    D 46H INCREMENT IN TIME STEP . . . . ( IELMB(3,I2,3H))= I5 /)
3000 FORMAT (1H1,21H ** STOP ** ERROR IN ,A10,13H BLOCK INPUT ,/
    1 14H FIRST STEP OF,I5,34HTH BLOCK IS LESS THAN LAST STEP OF,I5,
    1 9HTH BLOCK.  ///)
3001 FORMAT (1H1,20H ** STOP ** ERROR IN,A10,45HBLOCK INPUT. FINAL STEP
    1 OF LAST BLOCK INPUT =,I5,18H, LESS THAN NSTE =,I5)
3004 FORMAT (1H1,21H ** STOP ** ERROR IN ,A10,13H BLOCK INPUT,/
    1 14H FIRST STEP OF,I5,36HTH BLOCK IS LARGER THAN LAST STEP OF,I5,
    1 9HTH BLOCK.  ///)
3002 FORMAT (///45H **STOP - IMASS MUST BE GT.0 IF CONCENTRATED
    1       / 37H MASSES AND-OR DAMPERS ARE SPECIFIED      )
3003 FORMAT(///42H **STOP - IEIG.GT.1 NOT PERMITTED IN THIS
    *          19HVERSION OF ADINA**    )
3010 FORMAT (///,1H1,100H **STOP** IEIG.GT.0 NOT PERMITTED IF CENTRAL D
    1IFFERENCE METHOD IS TO BE USED FOR TIME INTEGRATION              )
3012 FORMAT (///,1H1,60H **STOP** IMASS MUST BE EQ. 1 FOR CENTRAL DIFFE
    1RENCE METHOD            )
3020 FORMAT (///,1H1,75H **STOP** IMASS MUST BE EQ. 1 FOR DYNAMIC ANALY
    1SIS INCLUDING GRAVITY LOADS       )
C
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
FORTRAN IV G1 RELEASE 2.0                ADINI              DATE = 81229

   *OPTIONS IN EFFECT*  NOTERM,ID,EBCDIC,SOURCE,NOLIST,NODECK,LOAD,NOMAP
   *OPTIONS IN EFFECT*  NAME = ADINI    , LINECNT =       60
   *STATISTICS*    SOURCE STATEMENTS =       355,PROGRAM SIZE =      22912
   *STATISTICS*  NO DIAGNOSTICS GENERATED
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
C *CDC* *DECK INLIST
C           DATA SET CLOADS      AT LEVEL TMP AS OF 08/17/81
C *CDC* *DECK CLOADS
      SUBROUTINE CLOADS (ID,RG,R,TIMV,RV,RMASS,MASS,NOD,NCUR,IDIRN,FAC,
     1                   ARTM,KL,RGST,NTFND,NDUF)
C
C     SUBROUTINE
C     1. TO READ THE TIME FUNCTIONS AND CALCULATE THE
C        FUNCTIONAL VALUES AT REQUESTED POINTS.
C     2. TO CALCULATE THE GRAVITY LOADING.
C     3. TO READ CONCENTRATED NODAL LOADS
C     4. TO CALCULATE THE LOAD VECTORS CORRESPONDING
C        TO THE CONCENTRATED LOADS.
C. . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C
C
C     V A R I A B L E S 0
C
C
C     ID       = ARRAY OF BOUNDARY CONDITION CODES
C     RG       = INTERPOLATED VALUES OF TIME FUNCTIONS
C     R        = LOAD VECTOR
C     TIMV,RV  = ABSCISSA AND ORDINATES OF TIME FUNCTIONS
C     NOD      = NODAL POINTS TO WHICH LOADS ARE APPLIED
C     NCUR     = TIME FUNCTION NUMBERS OF LOADS
C     IDIRN    = DIRECTION CODES OF LOADS
C     FAC      = MULTIPLIER OF LOADS
C     ARTM     = ARRIVAL TIMES OF LOADS
C     KL       = INCREMENTS IN NODES FOR GENERATION
C
C
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /SOL/ NUMNP,NEQ,NWK,NWM,NWC,NUMEST,MIDEST,MAXEST,NSTE,MA
      COMMON /CONST/ DT,DTA,A0,A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11
     1              ,A12,A13,A14,A15,A16,A17,A18,A19,A20,IOPE
      COMMON /VAR/ NG,MODEX,IUPDT,KSTEP,ITEMAX,IEQREF,ITE,KPRI,
     1            IREF,IEQUIT,IPRI,KPLOTN,KPLOTE
      COMMON /ADINAI/ OPVAR(7),TSTART,IRINT,ISTOTE
      COMMON /MDFROM/ IDOF(6)
      COMMON /LOA/ TEND,NTFN,NPTM,NLOAD,IDGRAV,NPR2,NPR3,NODE3
      COMMON /PRCON/ IDATWR,IPRIC,NPB,IDC,IVC,IAC,IPC,IPNODE(3,15)
      COMMON /BEAR/ IBEAR, MTOTB
C
      DIMENSION ID(NDOF,1),RG(NTFND,1),R(1),TIMV(1),RV(1),RMASS(1),
     1          NOD(1),IDIRN(1),NCUR(1),FAC(1),ARTM(1),KL(1)
     2          ,RGST(1),MASS(1)
      DIMENSION  THO(100),SP(100)
C-----------------------------------------------------------------
C-    THO(I)   -    INITIAL PHASE ANGLE OF ROTATING IMBALANCE LOAD   -
C-                                                                  -
C-                  APPLIED AT NODE I                               -
C-                                                                  -
C-    SP(I)    -    ROTATING SPEED OF IMBALANCE LOAD APPLIED AT      -
C-                                                                  -
C-                  NODE I                                          -
C-----------------------------------------------------------------
C
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
      REWIND 3
      REWIND 8
C
      READ (8) ((ID(I,J),I=1,NDOF),J=1,NUMNP)
C
C     CALCULATION OF TIME FUNCTION DATA AT ALL TIME POINTS
C
      IF (NSTE.EQ.0) RETURN
C
      IF (NTFN.GT.0) GO TO 60
C
      DO 40 I=1,NEQ
   40 R(I)=0.
      DO 50 K=1,NSTE
   50 WRITE (3) (R(I),I=1,NEQ)
      RETURN
C
   60 CALL TFUNCT (RG,TIMV,RV,RGST,NTFN,NSTE,MODEX,TSTART,DT,DTA)
C
C     CALCULATE GRAVITY LOADING
C
      NSTEG=0
C
      IF (IDGRAV.EQ.0) GO TO 100
C
      CALL GRAVL (ID,IDOF,MASS,RMASS,R,RG,NUMNP,NDOF,NEQ,NTFN,NSTEG,
     1            MODEX)
C
C     ADD CONCENTRATED LOADING
C
  100 IF (NSTEG.EQ.NSTE) RETURN
C
      IF (NLOAD.EQ.0) GO TO 120
      IF (IDATWR.LE.1.AND.IBEAR.EQ.1) WRI     2001)
      IF (IBEAR.EQ.1) GO TO 102
      IF (IDATWR.LE.1) WRITE (5,2000)
  102 CONTINUE
C
C     READ IMBALANCE LOAD INFORMATION : EMPLOYED ONLY FOR
C
C     ROTOR BEARING TYPE PROBLEMS
C
      IF (IBEAR.EQ.1) READ (5,1001) (NOD(I),IDIRN(I),NCUR(I),FAC(I),
     1                ARTM(I),KL(I),THO(I),SP(I),I=1,NLOAD)
      IF (IBEAR.EQ.1) GO TO 101
      READ (5,1000) (NOD(I),IDIRN(I),NCUR(I),FAC(I),ARTM(I),KL(I),
     1               IDEBU ,I=1,NLOAD)
  101 KL(NLOAD)=0
C
      DO 139 I=1,NLOAD
      IF (IDATWR.GT.1) GO TO 120
      IF (IBEAR.EQ.1) WRITE (6,2011)NOD(I),IDIRN(I),NCUR(I),FAC(I),
     1                ARTM(I),KL(I),THO(I),SP(I)
      IF (IBEAR.EQ.1) GO TO 139
  139 CONTINUE
C
  120 IF (MODEX.EQ.0) RETURN
C
```

```
      NLL=NSTEG + 1
      DO 200 K=NLL,NSTE
      DO 210 I=1,NEQ
  210 R(I)=0.
C
      IF (NLOAD.EQ.0) GO TO 260
C
      DO 220 L=1,NLOAD
      LI=IDIRN(L)
      IF (IDOF(LI).EQ.1) GO TO 220
      LDOF=LI
      LN=NOD(L)
      ARTMT=ARTM(L)
      FACT=FAC(L)
      LC=NCUR(L)
      IF (KL(L).EQ.0) GO TO 222
      DARTM=(ARTM(L+1) - ARTM(L))/((NOD(L+1) - NOD(L))/KL(L))
      FINCR=(FAC(L+1) - FAC(L))/((NOD(L+1) - NOD(L))/KL(L))
  222 DO 230 I=1,LDOF
  230 IF (IDOF(I).EQ.1) LI=LI - 1
  224 NSTEA=ARTMT/DT
      NSTEF=K - NSTEA
      IF (NSTEF.LE.0) GO TO 226
      AFACT=NSTEA - ARTMT/DT + 1.
C
      II=ID(LI,LN)
      IF (II.LE.0) GO TO 226
      RGFR=RG(LC,NSTEF)
      IF (ARTMT.EQ.0.) GO TO 240
C
      RGFR=RGST(LC)*(1.0 - AFACT) + RGFR*AFACT
      IF (NSTEF.LE.1) GO TO 240
      RGFR=RG(LC,NSTEF-1)*(1.0-AFACT)+RG(LC,NSTEF)*AFACT
  240 IF (IBEAR.NE.1) GO TO 245
C
C     ADD HORIZONTAL IMBALANCE LOAD COMPONENT TO EXTERNAL LOAD VECTOR
C
C     WRITE (6,1987) L,THO(L),SP(L)
 1987 FORMAT (5X,'L=',I10,5X,'THO(L)=',D13.5,5X,'SP(L)=',D13.5)
      R(II)=R(II)+RGFR*FACT*DCOS(THO(L)+SP(L)*DT*K)
C
C     ADD VERTICAL IMBALANCE LOAD COMPONENT TO EXTERNAL LOAD VECTOR
C
      R(II+1)=R(II+1)+RGFR*FACT*DSIN(THO(L)+SP(L)*DT*K)
      GO TO 246
  245 R(II)=R(II)+RGFR*FACT
  246 CONTINUE
  226 IF (KL(L).EQ.0) GO TO 220
      LN=LN + KL(L)
      IF (LN.GE.NOD(L+1)) GO TO 220
      FACT=FACT + FINCR
      ARTMT=ARTMT + DARTM
      GO TO 224
  220 CONTINUE
C
  260 IF (IDGRAV.EQ.0) GO TO 362
C
      DO 360 I=1,NEQ
```

CLOADS DATE = 81229 09/13/44

```
  360 R(I)=R(I) + RMASS(I)
  362 WRITE (3) (R(I),I=1,NEQ)
C     WRITE (6,3000) K,(R(I),I=1,NEQ)
 3000 FORMAT (5X,'TIME STEP-',I8,5X,'R(I)(CLOADS):',//,5X,8(3X,D12.5))
      IF ( IDEBUG.EQ.5) WRITE (6,6000) (R(I),I=1,NEQ)
  200 CONTINUE
C
      RETURN
 1000 FORMAT (3I5,2F10.0,I5,5X,I5)
 1001 FORMAT (3I5,2F10.0,I5,2F10.0)
 2000 FORMAT (////45H   C O N C E N T R A T E D   L O A D S   D A T A// 4X,
     1            53H NODE    DIRECTION     LOAD CURVE    LOAD CURVE MULTIPL     ,
     2            50H   ARRIVAL TIME    NODE GENERATION                    )
 2001 FORMAT (////'     B E A R I N G    C O N C E N T R A T E D    L O A
     1 D S    D A T A'///, 4X,' NODE    DIRECTION    LOAD CURVE    LOAD CURV
     2E MULTIPL    ARRIVAL TIME    NODE GENERATION    INITIAL ANGLE    LOAD
     3ROTATING SPEED')
 2010 FORMAT (1H0,2X,I5,5X,I4,9X,I4,9X,E13.5,8X,E12.4,7X,I5)
 2011 FORMAT (1H0,2X,I5,5X,I4,9X,I4,9X,E13.5,8X,E12.4,7X,I5,5X,
     1            E13.5,5X,E13.5)
 6000 FORMAT (10E12.5/)
      END
```

```
FORTRAN IV G1  RELEASE 2.0                CLOADS              DATE = 81229

   *OPTIONS IN EFFECT*  NOTERM,ID,EBCDIC,SOURCE,NOLIST,NODECK,LOAD,NOMAP
   *OPTIONS IN EFFECT*  NAME = CLOADS  , LINECNT =        60
   *STATISTICS*     SOURCE STATEMENTS =       101,PROGRAM SIZE =      6248
   *STATISTICS*  NO DIAGNOSTICS GENERATED
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
                    MAIN                          DATE = 81229          09/18/44
C          DATA SET COLSOL        AT LEVEL TMP AS OF 08/17/81
C
C
C
C *CDC* *DECK COLSOL
C *UNI* )FOR,IS   N.COLSOL, R.COLSOL
      SUBROUTINE COLSOL (MAXA,NCOLBV,ICOPL,A,B,D,V,NEQ,NBLOCK,ISTORL,
     1                   NSTIF,NRED,KKK)
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                     .
C .   P R O G R A M                                                     .
C .        TO SOLVE FINITE ELEMENT STATIC EQUILIBRIUM EQUATIONS OUT-OF .
C .        CORE, USING COMPACTED STORAGE AND COLUMN REDUCTION SCHEME   .
C .                                                                     .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /ELSTP/ TIME,IDTHF
      COMMON /EL/ IND,ICOUNT,NPAR(20),NUMEG,NEGL,NEGNL,IMASS,IDAMP,ISTAT
     1            ,NDOF,KLIN,IEIG,IMASSN,IDAMPN
      COMMON /RANDI/ NOA,N10,IELCPL
      DIMENSION A(ISTORL),B(ISTORL),D(NEQ),V(1)
      INTEGER ICOPL(1),NCOLBV(1),MAXA(1)
C
      KHBB=0
      IF (KKK-2) 10,610,610
   10 REWIND NSTIF
      PIVOT=10.**20
C
C --- FACTORIZE STIFFNESS MATRIX  ( LOOP OVER ALL BLOCKS ) ---
C
      DO 600 NJ=1,NBLOCK
C
      READ (NSTIF) A
C     WRITE (6,732) (A(I),I=1,ISTORL)
  732 FORMAT (5X,'STIFFNESS MATRIX (COLSOL):',//,5X,8(2X,D13.6))
      NCOLB=NCOLBV(NJ)
      MM=MAXA(KHBB+1) - 1
      IF (NJ.EQ.ICOPL(NJ)) GO TO 300
C
      IK=ICOPL(NJ) - 1
      IM=0
      IF (IK) 300,140,100
  100 DO 120 K=1,IK
  120 IM=IM + NCOLBV(K)
  140 KHB=KHBB - IM
      IK=IK + 1
      NJ1=NJ - 1
C
C     REDUCE BLOCK BY THE PRECEEDING COUPLING BLOCKS
C
      DO 160 NK=IK,NJ1
C
C
C       * * * * *           R A N D O M   A C C E S S         * * *
C
      NREC10=NK
      CALL READMS (NRED,B,ISTORL,NREC10)
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
C
C         * * * * *          R A N D O M   A C C E S S         * * *
C
      KHB=KHB - NCOLBV(NK)
      MC=MAXA(IM+1) - 1
C
      DO 200 N=1,NCOLB
      KN=MAXA(KHBB+N) - MM
      KL=KN + 1
      KU=MAXA(KHBB+N+1) - 1 - MM
      KH=KU - KL - N + 1
      KC=KH - KHB
      IF (KC.LE.0) GO TO 200
      IC=0
      KCL=NCOLBV(NK) - KC + 1
      IF (KCL.GT.0) GO TO 210
      IC=1 - KCL
      KCL=1
  210 KCR=NCOLBV(NK)
      KLT=KU - IC
C
      DO 220 K=KCL,KCR
      IC=IC + 1
      KLT=KLT - 1
      KI=MAXA(K+IM) - MC
      ND=MAXA(K+IM+1) - KI - MC - 1
      IF(ND) 220,220,230
  230 KK=MIN0(IC,ND)
      C=0.
      DO 240 L=1,KK
  240 C=C + B(KI+L)*A(KLT+L)
      A(KLT)=A(KLT) - C
  220 CONTINUE
  200 CONTINUE
C
      IM=IM + NCOLBV(NK)
C
  160 CONTINUE
C
C     REDUCE BLOCK BY ITSELF
C
  300 DO 400 N=1,NCOLB
      KN=MAXA(KHBB+N) - MM
      KL=KN + 1
      KU=MAXA(KHBB+N+1) - 1 - MM
      KDIF=KU - KL
      KH=MIN0(KDIF,N-1)
      KS=N + KHBB
      IF (KH) 420,440,460
  460 K=N - KH
      KLT=KL + KH
      IC=0
      IF ((N-1).LT.KDIF) IC=KDIF - N + 1
C
      DO 480 J=1,KH
      IC=IC + 1
      KLT=KLT - 1
      KI=MAXA(KHBB+K) - MM
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
      ND=MAXA(KHBB+K+1) - KI - MM - 1
      IF (ND) 480,480,500
  500 KK=MINO(IC,ND)
      C=0.
      DO 520 L=1,KK
  520 C=C + A(KI+L)*A(KLT+L)
      A(KLT)=A(KLT) - C
  480 K=K + 1
C
  440 K=KS
      E=0.
      DO 540 KK=KL,KU
      K=K - 1
      C=A(KK)/D(K)
      E=E + C*A(KK)
  540 A(KK)=C
      A(KN)=A(KN) - E
C
  420 D(KS)=A(KN)
      IF (D(KS)) 401,555,400
  555 IF (IDTHF.EQ.0) GO TO 560
      D(KS)=PIVOT
      GO TO 400
  560 WRITE (6,2000) KS,D(KS)
      STOP
C
  401 WRITE(6,2000) KS,D(KS)
  400 CONTINUE
C
      KHBB=KHBB + NCOLB
C
C        * * * * * *          R A N D O M   A C C E S S        * * *
C
      NREC10=NJ
      CALL WRITMS (NRED,A,ISTORL,NREC10)
C
C        * * * * * *          R A N D O M   A C C E S S        * * *
C
  600 CONTINUE
      IF (KLIN.GT.0) GO TO 606
      RETURN
C
C - - SOLUTION OF EQUATIONS ( LOOP OVER ALL BLOCKS ) - -
C
C     REDUCE THE LOAD VECTOR
C
  606 KHBB=0
  610 DO 700 NJ=1,NBLOCK
      IF (NBLOCK.EQ.1 .AND. (KLIN.EQ.0 .OR. KKK.EQ.1) ) GO TO 710
C
C        * * * * * *          R A N D O M   A C C E S S        * * *
C
      NREC10=NJ
      CALL READMS (NRED,A,ISTORL,NREC10)
C
C        * * * * * *          R A N D O M   A C C E S S        * * *
C
  710 NCOLB=NCOLBV(NJ)
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
      MM=MAXA(KHBB+1) - 1
      DO 720 N=1,NCOLB
      KL=MAXA(N+KHBB) - MM + 1
      KU=MAXA(N+KHBB+1) - MM - 1
      IF (KU-KL) 720,730,730
  730 KS=N + KHBB
      K=KS
      C=0.
      DO 740 KK=KL,KU
      K=K - 1
  740 C=C + A(KK)*V(K)
      V(KS)=V(KS) - C
  720 CONTINUE
      KHBB=KHBB + NCOLB
  700 CONTINUE
C
C     BACKSUBSTITUTE
C
      DO 790 N=1,NEQ
  790 V(N)=V(N)/D(N)
      NBL=NBLOCK
      DO 800 NJ=1,NBLOCK
      IF (NBLOCK.EQ.1) GO TO 820
C
C        * * * * * *        R A N D O M   A C C E S S        * * *
C
      NJB1=NBLOCK - NJ + 1
      CALL READMS (NRED,A,ISTORL,NJB1)
C
C        * * * * * *        R A N D O M   A C C E S S        * * *
C
      NCOLB=NCOLBV(NBL)
  820 KHBB=KHBB - NCOLB
      MM=MAXA(KHBB+1) - 1
      N=NCOLB
      DO 860 L=1,NCOLB
      KL=MAXA(N+KHBB) - MM + 1
      KU=MAXA(N+KHBB+1) - MM - 1
      IF (KU-KL) 861,890,890
  890 KS=KHBB + N
      K=KS
      DO 900 KK=KL,KU
      K=K - 1
  900 V(K)=V(K) - A(KK)*V(KS)
  861 N=N-1
  860 CONTINUE
      NBL=NBL - 1
  800 CONTINUE
C
      RETURN
 2000 FORMAT (// 40H STOP - STIFFNESS NOT POSITIVE DEFINITE  ,//
     1           32H NONPOSITIVE PIVOT FOR EQUATION  ,I4,//
     2           10H PIVOT =  ,E20.12 )
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
FORTRAN IV G1   RELEASE 2.0                COLSUL                DATE = 81229

   *OPTIONS IN EFFECT*  NOTERM,ID,EBCDIC,SOURCE,NOLIST,NODECK,LOAD,NOMAP
   *OPTIONS IN EFFECT*  NAME = COLSOL  , LINECNT =       60
   *STATISTICS*     SOURCE STATEMENTS =       154,PROGRAM SIZE =      4280
   *STATISTICS*  NO DIAGNOSTICS GENERATED
```

```
                 MAIN                    DATE = 81245            15/23/01

C           DATA SET RUSS          AT LEVEL TMP AS OF 09/02/81
C
C
C  *CDC*  *DECK RUSS
C  *UNI*  )FOR,IS   N,RUSS,    R,RUSS
      SUBROUTINE RUSS (ID,X,Y,Z,HT,E,DEN,AREA,LM,XYZ,MATP,EPSIN,
     1                 IPS,ETIMV,EDISB,WA,PROP,NODGL,IELTD,NCON,
     2                 NODE,NDM,IDWA,TEMPV1,TEMPV2,MXNODS)
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /SOL/ NUMNP,NEQ,NWK,NWM,NWC,NUMEST,MIDEST,MAXEST,NSTE,MA
      COMMON /EL/ IND,ICOUNT,NPAR(20),NUMEG,NEGL,NEGNL,IMASS,IDAMP,ISTAT
     1           ,NODFDM,KLIN,IEIG,IMASSN,IDAMPN
      COMMON /DIM/ N0,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /CONST/ DT,DTA,A0,A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11
     1              ,A12,A13,A14,A15,A16,A17,A18,A19,A20,IOPE
      COMMON /ELSTP/ TIME,IDTHF
      COMMON /EM1D/ S(78),XM(24),ST(6),D(4),RE(24)
      COMMON /VAR/ NG,MODEX,IUPDT,KSTEP,ITEMAX,IEQREF,IIE,KPRI,
     1             IREF,IEQUIT,IPRI,KPLOTN,KPLOTE
      COMMON /PRCON/ ICATWR,IPRIC,NPB,IDC,IVC,IAC,IPC,IPNODE(3,15)
      COMMON /PORT/ INPORT,JNPORT,NPUTSV,LUNODE,LU1,LU2,LU3,JDC,JVC,JAC
      COMMON /MDFRDM/ IDOF(6)
      COMMON /RANDI/ NOA,NID,IELCPL
      COMMON /GAUSS/ XG(4,4),WGT(4,4),EVAL2(9,2),EVAL3(27,3),E1,E2,E3
      COMMON /TRNODE/ RST(12),DISP(12),PP(4),STS(4),STN(4),RL(4),IELD,ND
      COMMON /TMODEL/ LINEL,NONEL,ITEL,ISPEL,KINPEL,IEPCL,IEPCK,MODMAX
      COMMON /DIMEE/ NB,NB1,NB2,NB3,NB4,NB5,NB6,NB7,NB8,NB9,NB10,NB11,

     1             NB12,NB13,NB14,NB15
C
      COMMON /BEAR/ IBEAR,MTOTB
      COMMON /BEAR2/ B(1)
      COMMON /DPR/ ITWO
      COMMON /PLOTB/ MB,NR(100),NS(100)
      COMMON A(1)
      REAL A
      REAL B
      DIMENSION VELB(12),ACCB(12),IPLOTN(100)

      DIMENSION ID(NODF,1),X(1),Y(1),Z(1),HT(1),E(1),DEN(1),AREA(1),
     1          LM(NDM,1),XYZ(NDM,1),MATP(1),EPSIN(1),IPS(1),ETIMV(1),
     2    EDISB(NDM,1),WA(IDWA,1),PROP(NCON,1),
     3          NODGL(MXNODS,1),IELTD(1),TEMPV1(1),TEMPV2(1)
C
      DIMENSION EC(12),NODE(4),NODEM(4),PM(4)
      DIMENSION ICOFTB(12)
      DATA RECLB1/8HMATERAL1/, RECLB2/8HELEMENT1/, RECLB3/8HNEWSTEP1/,
     1     RECLB4/8HOUTPUT=1/
      DATA HEAD1/6HRADIUS/, HEAD2/6HLENGTH/
C
      EQUIVALENCE (NPAR(1),NPAR1), (NPAR(2),NUMEL), (NPAR(3),INONL),
     1            (NPAR(4),IDEATH), (NPAR(5),ITYPT), (NPAR(10),NINT),
     2            (NPAR(15),MODEL), (NPAR(16),NUMMAT)
C
C   ** NOTE ** DURING THE TIME INTEGRATION, X=DISP, Y=VEL, Z=ACC
C
      IELCPL=0
```

```
      IF (KPRI.EQ.0) GO TO 800
      IF (IND.GT.0) GO TO 420
C
C     SKIP TO BEARING CALCULATIONS IF NECESSARY
C
      IF (IBEAR.EQ.1.AND.ITYPT.EQ.3) GO TO 80
C
C
C
C  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C  .  R E A D   A N D   G E N E R A T E   E L E M E N T              .
C  .  I N F O R M A T I O N                                            .
C  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C
C
C
C         1. READ MATERIAL PROPERTIES
C
C
      IF (IDATWR.LE.1) WRITE (6,2000)
C
      GO TO (10,20,30,40,40,60,60,70), MODEL
C
C     LINEAR ELASTIC  (MODEL 1)
C
   10 IF (IDATWR.LE.1) WRITE (6,2010)
      DO 15 I=1,NUMMAT
      READ (5,1000) N,AREA(N),DEN(N)
      READ (5,1010) E(N)
      IF (IDATWR.LE.1) WRITE (6,2011) N,AREA(N),DEN(N),E(N)
   15 CONTINUE
C
C
C***  DATA PORTHOLE  ***************************** (START)
C
      IF (JNPCRT.EQ.0 .OR. NPUTSV.EQ.0) GO TO 150
      RECLAB = RECLB1
      WRITE (LU1) RECLAB,NUMMAT,NCON,(DEN(I),I=1,NUMMAT),
     1             (E(I),I=1,NUMMAT),(AREA(I),I=1,NUMMAT)
C
C***  DATA PORTHOLE  ***************************** ( END )
C
      GO TO 150
C
C     NONLINEAR ELASTIC  (MODEL 2)
C
   20 IP=NCON/2
      KP=1
      DO 25 I=1,NUMMAT
      READ (5,1000) N,AREA(N),DEN(N)
      READ (5,1010) (PROP(J,N),J=1,NCON)
      IF (IDATWR.GT.1) GO TO 25
      WRITE (6,2020) N,AREA(N),DEN(N),KP,PROP(1,N),PROP(IP+1,N)
      DO 22 K=2,IP
      KP=K+IP
      ETAN=(PROP(KP,N) - PROP(KP-1,N))/(PROP(K,N) - PROP(K-1,N))
   22 WRITE (6,2021) K,PROP(K,N),PROP(KP,N),ETAN
```

```
   25 CONTINUE
      GO TO 98
C
C     THERMOELASTIC  (MODEL 3)
C
   30 DO 35 I=1,NUMMAT
      READ (5,1000) N,AREA(N),DEN(N)
      READ (5,1010) (PROP(J,N),J=1,NCON)
      IF (IDATWR.GT.1) GO TO 35
      WRITE (6,2030) N,AREA(N),DEN(N)
      WRITE (6,2031) (PROP(J,N),J=1,NCON)
   35 CONTINUE
      GO TO 98
C
C     ELASTIC-PLASTIC MODELS  (MODEL 4  AND  MODEL 5)
C
   40 IF (IDATWR.LE.1) WRITE (6,2040)
      DO 45 I=1,NUMMAT
      READ (5,1000) N,AREA(N),DEN(N)
      READ (5,1010) (PROP(J,N),J=1,NCON)
      IF (IDATWR.GT.1) GO TO 45
      WRITE (6,2011) N,AREA(N),DEN(N),(PROP(J, N),J=1,NCON)
   45 CONTINUE
      GO TO 98
C
C     THERMO-ELASTIC-PLASTIC AND CREEP MODELS (MODEL 6 AND MODEL 7)
C
   60 DO 65 I=1,NUMMAT
      READ (5,1000) N,AREA(N),DEN(N)
      READ (5,1010) (PROP(J,N),J=1,NCON)
      IF (IDATWR.GT.1) GO TO 65
      WRITE (6,2030) N,AREA(N),DEN(N)
      WRITE (6,2061) (PROP(J,N),J=1,NCON)
   65 CONTINUE
      GO TO 98
C
C
C     USER-SUPPLIED MODEL  ( MODEL 8 )
C
   70 DO 75 I=1,NUMMAT
      READ (5,1000) N,AREA(N),DEN(N)
      READ (5,1010) (PROP(J,N),J=1,NCON)
      IF (IDATWR.GT.1) GO TO 75
      WRITE (6,2030) N,AREA(N),DEN(N)
      WRITE (6,2071) (J,N,PROP(J,N),J=1,NCON)
   75 CONTINUE
C
C
C***  DATA PORTHOLE  **************************** (START)
C
   98 IF (JNPORT.EQ.0 .OR. NPUTSV.EQ.0) GO TO 150
      RECLAB = RECLB1
      WRITE (LU1)  RECLAB,NUMMAT,NCON,(DEN(I),I=1,NUMMAT),
     1             ((PROP(I,J),I=1,NCON),J=1,NUMMAT),
     2             (AREA(I),I=1,NUMMAT)
C
C***  DATA PORTHOLE  **************************** ( END )
      GO TO 150
```

```
C**************************************************************************
C*                                                                       *
C*     DYNAMIC ALLOCATION STORAGE FOR BEARING DATA                       *
C*                                                                       *
C*      NB     NOMINAL DAMPER ANNULUS DIAMETER(BD)                       *
C*                                                                       *
C*      NB1    NOMINAL DAMPER ANNULUS LENGTH(BL)                         *
C*                                                                       *
C*      NB2    DAMPER ANNULUS RADIAL CLEARENCE(BC)                       *
C*                                                                       *
C*      NB3    DAMPER LUBRICANT VISCOSITY(VISC)                          *
C*                                                                       *
C*      NB4    FILM RUPTURE PRESSURE(PVAP)                               *
C*                                                                       *
C*      NB5    POSITION ANGLE OF LUBRICANT PORT-1(TH1)                   *
C*                                                                       *
C*      NB6    POSITION ANGLE OF LUBRICANT PORT-2(TH2)                   *
C*                                                                       *
C*      NB7    SPECIFIED BOUNDARY PRESSURE AT PORT-1(PB1)                *
C*                                                                       *
C*      NB8    SPECIFIED BOUNDARY PRESSURE AT PORT-2(PB2)                *
C*                                                                       *
C*      NB9    NUMBER OF FINITE-DIFFERENCE GRID POINTS                   *
C*                                                                       *
C*               PER DAMPER ARC(NGRID)                                   *
C*                                                                       *
C*      NB10   BEARING SOLUTION OPTION(NSOLN):                           *
C*                                                                       *
C*             NSOLN=1     LONG-BEARING SOLUTION USED                    *
C*                                                                       *
C*             NSOLN=2     SHORT-BEARING SOLUTION USED                   *
C*                                                                       *
C*             NSOLN=3     FOURIER-SERIES  2-D                           *
C*                                                                       *
C*      NB11   NUMBER OF LUBRICANT PORTS (NPORTS) (0,1,2)                *
C*                                                                       *
C*             IF NPORT=0, JOINED-BOUNDARY CONDITION IS USED             *
C*                                                                       *
C*      NB12   NUMBER OF IDENTICAL ANNULI FOR THE DAMPER(NELM)           *
C*                                                                       *
C*      NB13   STIFFNESS MATRIX OPTION (KOFK):                           *
C*                                                                       *
C*             KOFK=0  STIFFNESS NOT COMPUTED                            *
C*                                                                       *
C*             KOFK=1  STIFFNESS MATRIX COMPUTED                         *
C*                                                                       *
C*      NB14   DAMPING MATRIX OPTION(KOFC):                              *
C*                                                                       *
C*             KOFC=0  DAMPING MATRIX NOT COMPUTED                       *
C*                                                                       *
C*             KOFC=1  DAMPING MATRIX COMPUTED                           *
C*                                                                       *
C**************************************************************************
C
C
   80 NB    = 1
C
      NB1   = NB    + NUME*ITWO
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
      NB2  = NB1  + NUME*ITWO
      NB3  = NB2  + NUME*ITWO
      NB4  = NB3  + NUME*ITWO
      NB5  = NB4  + NUME*ITWO
      NB6  = NB5  + NUME*ITWO
      NB7  = NB6  + NUME*ITWO
      NB8  = NB7  + NUME*ITWO
      NB9  = NB8  + NUME*ITWO
      NB10 = NB9  + NUME
      NB11 = NB10 + NUME
      NB12 = NB11 + NUME
      NB13 = NB12 + NUME
      NB14 = NB13 + NUME
      NB15 = NB14 + NUME
C
C     CHECK IF DYNAMIC ARRAY B HAS ENOUGH ROOM
C
C     FOR THE BEARING ELEMENT INFORMATION
C
      CALL SIZEB(NB15)
C
C     SET THE NUMBER OF ELEMENTS OF THE BEARING ELEMENT GROUP.
C
C      CALL SUBROUTINE BEARNG TO READ BEARING ELEMENT INFORMATION
C
      CALL BEARNG(B(NB),B(NB1),B(NB2),B(NB3),B(NB4),B(NB5),
C
     1            B(NB6),B(NB7),B(NB8),B(NB9),B(NB10),B(NB11),
C
     2            B(NB12),B(NB13),B(NB14),A(N3),IELTD,X,Y,Z,XYZ,LM,
C
     3            S,RE,MADR,ISTIFB,NDM,NUME)
C
C
C
C
C
C
C         2. READ ELEMENT INFORMATION
C
  150 HEAD=HEAD1
      IF (ITYPT.EQ.0) HEAD=HEAD2
      IF (IDATWR.LE.1.AND.ITYPT.NE.3) WRITE (6,2200) HEAD
      IF (IDATWR.LE.1.AND.ITYPT.EQ.3) WRITE (6,2201)
      MB=0
      N=1
C
      IREAD=5
      IF (INPCRT.GT.0) IREAD=59
  160 IF (ITYPT.EQ.3) READ(IREAD,1103)M, IELE, KG,
     1                (IDOFTB(IB),IB=1,12),IPLOTE
      IF (ITYPT.NE.3) READ (IREAD,1100) M, IELE, IS, MTYP, KG, EPS,
     1                                  ETIME
 1103 FORMAT (16I4)
      READ (IREAD,1200) NODE
C
      IF (IDEATH.EQ.2 .AND. ETIME.EQ.0.) ETIME=100000.
      IF (IELE.EQ.0) IELE=MXNODS
```

```
      IF (IBEAR.EQ.1.AND.ITYPT.EQ.3) GO TO 120
      IF (MTYP.LE.0) MTYP=1
      IF (MTYP.GT.NUMMAT) GO TO 110
      IF (KG.LE.0) KG=1
      IF (IELE.LE.MXNODS) GO TO 120
C
      WRITE (6,2300) NG,M,IELE,MXNODS
      STOP
C
  110 WRITE (6,2310) NG,M,MTYP,NUMMAT
      STOP
C
  120 IF(M.NE.N) GO TO 200
      DO 122 I=1,IELE
  122 NODEM(I)=NODE(I)
      MTYPE=MTYP
      KKK=KG
      EPSI=EPS
      IPST=IS
      ETIM=ETIME
      IELD=IELE
      IPLOT=IPLOTE
C
C     SAVE ELEMENT INFORMATION
C
  200 IF (ITYPT.NE.1) GO TO 195
      I=NODEM(1)
      XYZ(1,N)=Y(I)
      GO TO 201
C
  195 L=-2
      DO 190 LL=1,IELD
      L=L + 3
      I=NODEM(LL)
      XYZ(L,N)=X(I)
      XYZ(L+1,N)=Y(I)
  190 XYZ(L+2,N)=Z(I)
C
C
  201 MATP(N)=MTYPE
      EPSIN(N)=EPSI
      IPS(N)=IPST
      IELTC(N)=IELD
      IPLOTN(N)=IPLOT
      IF (ITYPT.EQ.3.AND.IPLOT.EQ.1) MB=MB+1
      IF (ITYPT.EQ.3.AND.IPLOT.EQ.1) NR(MB)=NODEM(1)
      IF (ITYPT.EQ.3.AND.IPLOT.EQ.1) NS(MB)=NODEM(2)
      ND=3*IELD
      IF (ITYPT.EQ.1) ND=1
      IF (IBEAR.EQ.1.AND.ITYPT.EQ.3) ND=12
      IF (ITYPT.EQ.3) GO TO 204
C
C     INITIALIZE WORKING STORAGE
C
      IF(MODEL.LT.3) GO TO 204
C
      CALL IMATB (PROP(1,MTYPE),WA(1,N),WA(1,N),NODGL(1,N),
     1            TEMPV1,NODEM,IELD)
```

```
C
  204 IF (IDEATH.EQ.0) GO TO 210
      IF (IDEATH.EQ.2) GO TO 207
      DO 208 L=1,ND
  208 EDISB(L,N)=0.
      ETIMV(N)=-ETIM
      GO TO 210
  207 ETIMV(N)=ETIM
C
  210 IF (ITYPT.NE.1) GO TO 211
      LDO=1
      IF (IDOF(1).EQ.0) LDO=2
      LM(1,N)=0
      IF (IDOF(2).EQ.1) GO TO 295
      II=NODEM(1)
      LM(1,N)=ID(LDO,II)
      GO TO 295
C
  211 DO 290 L=1,ND
  290 LM(L,N)=0
      IF (IBEAR.EQ.1.AND.ITYPT.EQ.3) GO TO 230
      LL=1
      DO 240 L=1,3
      IF (IDOF(L).EQ.1) GO TO 240
      LP=L - 3
      DO 241 LK=1,IELD
      LP=LP + 3
      II=NODEM(LK)
      LM(LP,N)=ID(LL,II)
C     WRITE (6,9887) N,L,LL,LP,II,IDOF(L)
 9887 FORMAT (5X,'N=',I5,5X,'L=',I5,5X,'LL=',I5,5X,'LP=',I5,
     15X,'II=',I5,5X,'IDOF(L)=',I5)
  241 CONTINUE
      LL=LL + 1
  240 CONTINUE
C     WRITE (6,5098) ((ID(KD,MD),KD=1,6),MD=1,7)
C     WRITE (6,9786) (LM(III,N),III=1,ND)
 5098 FORMAT (5X,'ID(I,J) (TRUSS ELE)=',/,8(5X,I10))
  295 CONTINUE
C
      GO TO 233
C
C     CREATE THE BEARING ELEMENT CONNECTIVITY VECTOR LM
C
  230 LL = 1
      DO 232 L=1,6
      IF(IDOF(L).EQ.1) GO TO 232
      LP = L-6
      DO 231 LK=1,IELD
      LP = LP + 6
      II = NODEM(LK)
      LM(LP,N) = ID(LL,II)
      IF (IDOFTB(LP).EQ.1) LM(LP,N)=0
C     WRITE (6,9887) N,L,LL,LP,II,IDOF(L)
  231 CONTINUE
      LL=LL+1
  232 CONTINUE
C     WRITE (6,3526) NODE
```

```
 3526 FORMAT ('$$$$$$$$$$$$$$$$$$$$$$$$$  NDOF=',I5)
C      WRITE (6,5098) ((ID(KD,MD),KD=1,6),MD=1,7)
C      WRITE (6,6667) (NODEM(LK),LK=1,IELD)
 6667 FORMAT(5X,'NODEM(LK)=',2I20)
C      DO 239 LK=1,IELD
C      LI = NODEM(LK)
C      WRITE( 6,4369) (ID(LL,LI),LL=1,6)
  239 CONTINUE
 4369 FORMAT(5X,'ID(LL,II)=',6I12)
C      WRITE (6,9786) (LM(III,N),III=1,ND)
 9786 FORMAT (8X,'LM=',8(1X,I11))
C      UPDATE COLUMN HEIGHTS AND BANDWIDTH
C
  233 CALL COLHT (HT,ND,LM(1,N))
C
      IF (IDATWR.GT.1) GO TO 298
      RL(1)=XYZ(1,N)
      IF (ITYPT.EQ.0) CALL LENTH1 (XYZ(1,N),RL)
      IF (ITYPT.NE.3) WRITE (6,2210) N,IELTD(N),IPS(N),MATP(N),KKK,
     1                EPSIN(N),RL(1),ETIM,(NODEM(LL),LL=1,IELD)
      IF (ITYPT.EQ.3) WRITE (6,2211) N,IELTD(N),KKK,ETIM,
     1                (NODEM(LL),LL=1,IELD)
C
C
C***  DATA PORTHOLE  *************************** (START)
C
  298 IF (JNPORT.EQ.0 .OR. NPUTSV.EQ.0) GO TO 300
      RECLAB=RECLB2
      WRITE (LU1) RECLAB,N,IELTD(N),IPS(N),MATP(N),KKK,EPSIN(N),ETIM,
     1            (NODEM(LL),LL=1,IELD)
C
C***  DATA PORTHOLE  *************************** ( END )
C
C
  300 IF (N.EQ.NUME) RETURN
      N=N+1
      DO 220 LL=1,IELD
  220 NODEM(LL)=NODEM(LL) + KKK
      IF (N.GT.M) GO TO 100
      GO TO 120
C
  420 IF(IBEAR.EQ.1.AND.ITYPT.EQ.3) GO TO 720
      GO TO (440,560,560,700),IND
C
C
C     A S S E M B L E   L I N E A R   S T I F F N E S S   M A T R I C E S
C
C
  440 DO 500 N=1,NUME
      ILLD=IELTD(N)
      ND=IELD*3
      IF (ITYPT.EQ.1) ND=1
      CALL ECHECK (LM(1,N),ND,ICODE,IUPDT)
      IF (ICODE.EQ.1) GO TO 500
      MTYPE=MATP(N)
      IF (ITYPT.NE.1) GO TO 501
C
C     RING ELEMENT
```

```
C
      ND=1
      S(1)=E(MTYPE)*AREA(MTYPE)/XYZ(1,N)
      GO TO 520
C
C     2-4 NODE TRUSS
C
  501 AE=AREA(MTYPE)*E(MTYPE)
      CALL LENTH1 (XYZ(1,N),RL)
      CALL STIF1 (XYZ(1,N),AE,S)
C     WRITE (6,1113) (S(L),I=1,78)
C     WRITE (6,1114) (LM(I,N),I=1,6)
 1113 FORMAT(5X,'TRUSS STIFFNESS (RUSS):',//,8(2X,D13.5))
 1114 FORMAT (5X,'LM(I,N) (RUSS):',//,8(3X,I10))
C
  520 CALL ADDBAN (A(N2),A(N1),S,RE,LM(1,N),ND,1)
  500 CONTINUE
      RETURN
C
C
C     A S S E M B L E   M A S S   M A T R I C E S
C
C
  600 IF (IMASS.EQ.2) GO TO 550
C
C     LUMPED MASS DISCRETIZATION
C
  543 DO 640 N=1,NUME
      MTYPE=MATP(N)
      IELD=IELTD(N)
      ND=IELD*3
C
      IF (ITYPT.NE.1) GO TO 546
      XM(1)=XYZ(1,N)*AREA(MTYPE)*DEN(MTYPE)
      ND=1
      GO TO 547
C
  546 CALL LENTH1 (XYZ(1,N),RL)
C
      PM(1)=.5*RL(2)
      GO TO (900,620,625,626), IELD
  620 PM(2)=PM(1)
      GO TO 627
  625 PM(2)=.5*(RL(1)-RL(2))
      PM(3)=.5*RL(1)
      GO TO 627
  626 PM(2)=.5*(RL(1)-RL(3))
      PM(3)=.5*RL(3)
      PM(4)=.5*(RL(1)-RL(2))
C
  627 K=0
      ADEN=AREA(MTYPE)*DEN(MTYPE)
      DO 628 I=1,IELD
      XMM=ADEN*PM(I)
      DO 628 L=1,3
      K=K+1
  628 XM(K)=XMM
C
```

```
  547 CALL ADDMA (A(N4),XM,LM(1,N),ND)
  640 CONTINUE
C
      RETURN
C
C     CONSISTENT MASS DISCRETIZATION
C
  550 DO 650 N=1,NUME
      IELD=IELTD(N)
      ND=IELD*3
      IF (ITYPT.EQ.1) ND=1
      CALL ECHECK(LM(1,N),ND,ICODE,IUPDT)
      IF (ICODE.EQ.1) GO TO 650
      MTYPE=MATP(N)
C
      IF (ITYPT.NE.1) GO TO 651
      XM(1)=XYZ(1,N)*AREA(MTYPE)*DEN(MTYPE)
      ND=1
      GO TO 655
C
  651 ADEN=AREA(MTYPE)*DEN(MTYPE)
      CALL LENTH1 (XYZ(1,N),RL)
      CALL MASBAR (XYZ(1,N),ADEN,S)
C
  655 CALL ADDBAN (A(N2),A(N1),S,RE,LM(1,N),ND,1)
  650 CONTINUE
C
      RETURN
C
C
C     A S S E M B L E   N O N L I N E A R   F I N A L   S T R U C -
C     T U R E   S T I F F N E S S   A N D   E F F E C T I V E   L O A D
C     V E C T O R S
C
  700 ISTIF=0
      IF (ICOUNT.EQ.3) GO TO 703
      IF (IREF.EQ.0) ISTIF=1
  703 CONTINUE
      MADR=N3
      IF (ICOUNT.EQ.3) MADR=N5
C
      DO 710 N=1,NUME
      IELD=IELTD(N)
      ND=IELD*3
      IF (ITYPT.EQ.1) ND=1
      CALL ECHECK (LM(1,N),ND,ICODE,IUPDT)
      IF (ICODE.EQ.1) IELCPL=IELCPL + 1
      IF (ICODE.EQ.1) GO TO 710
      IF (IDEATH.EQ.0) GO TO 692
      ETIM=DABS(ETIMV(N))
      IF (IDEATH.EQ.2) GO TO 690
      IF (TIME.LT.ETIM) GO TO 710
      IF (ETIMV(N).GE.0.) GO TO 692
      ETIMV(N)=ETIM
      DO 695 L=1,ND
      I=LM(L,N)
      IF (I.LE.0) GO TO 695
      EDISE(L,N)=X(I)
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
  695 CONTINUE
      GO TO 692
  690 IF (TIME.GT.ETIM) GO TO 710
  692 MTYPE=MATP(N)
C
      IF (ITYPT.NE.1) GO TO 701
      ND=1
      RST(1)=XYZ(1,N)
      DISP(1)=0.
      I=LM(1,N)
      IF (I.GT.0) DISP(1)=X(I)
      IF (IDEATH.NE.1) GO TO 706
      RST(1)=RST(1) + EDISB(1,N)
      DISP(1)=DISP(1) - EDISB(1,N)
      GO TO 706
C
  701 DO 702 L=1,ND
      RST(L)=XYZ(L,N)
      DISP(L)=0.
      I=LM(L,N)
      IF (I.GT.0) DISP(L)=X(I)
      IF (IDEATH.NE.1) GO TO 702
      RST(L)=RST(L) + EDISB(L,N)
      DISP(L)=DISP(L) - EDISB(L,N)
  702 CONTINUE
C
C
  706 CALL STIFN1 (N,AREA(MTYPE),PROP(1,MTYPE),WA(1,N),
     1              NODGL(1,N),TEMPV1,TEMPV2,EPSIN(N),S,RE,ISTIF)
C
C
C
      CALL ADDBAN (A(MADR),A(N1),S,RE,LM(1,N),ND,2)
      IF (ISTIF.EQ.0) GO TO 710
C
C     ADD ELEMENT STIFFNESS
C
      CALL ADDBAN (A(N4),A(N1),S,RE,LM(1,N),ND,1)
C
  710 CONTINUE
      IF (IELCPL.EQ.NUME) IELCPL=-1
C
      RETURN
C
C.......................................................................
C.                                                                     .
C.     CALCULATE AND ASSEMBLE INSTANTANEOUS FORCE VECTOR               .
C.                                                                     .
C.     AND ITS SPATIAL GRADIENTS, I.E. THE TANGENT STIFFNESS           .
C.                                                                     .
C.     AND DAMPING MATRICES FOR THE NONLINEAR TIME-TRANSIENT           .
C.                                                                     .
C.     SQUEEZE-FILM DAMPER INTERACTIVE ELEMENT                         .
C.                                                                     .
C.......................................................................
  720 IF (IREF.EQ.0) ISTIFB=1
      MADR = N3
      IF(ICOUNT.EQ.3) MADR=N5
```

```
C
C     CALL SUBROUTINE BEARNG TO CALCULATE BEARING ELEMENT FORCES
C
C     AND STIFFNESSES
C
C     WRITE(6,1314) NEQ
 1314 FORMAT(5X,'*******%%%%%%%%%%*******************%%%%%%%%%%%*********',
     1  10X 'NEQ=',I10)
C     WRITE(6,1315) (X(I),I=1,NEQ)
 1315 FORMAT(5X,'X=',6E12.5)
C     WRITE(6,1316) (Y(I),I=1,NEQ)
 1316 FORMAT(5X,'Y=',6E12.5)
C     WRITE(6,1317) (Z(I),I=1,NEQ)
 1317 FORMAT(5X,'Z=',6E12.5)
      CALL BEARNG(B(NB),B(NB1),B(NB2),B(NB3),B(NB4),
C
     1                 B(NB5),B(NB6),B(NB7),B(NB8),B(NB9),
C
     2                 B(NB10),B(NB11),B(NB12),B(NB13),B(NB14),A(N3),
C
     3                 IELTD,X,Y,Z,XYZ,LM,S,RE,MADR,ISTIFB,NUM,NUME)
C
C
C
      RETURN
C
C
C     S T R E S S   C A L C U L A T I O N S
C
C
C
C***  DATA PORTHOLE  ***************************** (START)
C
  800 IF (JNPORT.EQ.0 .OR. KPLOTE.NE.3) GO TO 802
      RECLAB = RECLB3
      WRITE (LU1)  RECLAB,NG,(NPAR(I),I=1,20),KSTEP,TIME,NEGL
C
C***  DATA PORTHOLE  ***************************** ( END )
C
  802 IF (NG.GT.NEGL) GO TO 805
C
C     GEOMETRIC AND MATERIAL LINEAR STRESS CALCULATION
C
      IPRNT=0
      IPORT=JNPORT*KPLOTE
C
      DO 830 N=1,NUME
      IPST=IPS(N)
      IF (IPST.EQ.0) GO TO 830
      IF (IPRI.NE.0) GO TO 801
      IPRNT=IPRNT + 1
      IF (IPRNT.NE.1) GO TO 801
      WRITE (6,2500) NG
  801 MTYPE=MATP(N)
      IELD=IELTD(N)
      ND=IELD*3
C
      IF (ITYPT.NE.1) GO TO 811
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
      ND=1
      EP=EPSIN(N)
      I=LM(1,N)
      IF (I.GT.0) EP=EP + X(I)/XYZ(1,N)
      STN(1)=EP
      STS(1)=E(MTYPE)*EP
      PP(1)=AREA(MTYPE)*STS(1)
      IF (IPRI.EQ.0) WRITE(6,2510) N,ND,PP(1),STS(1),STN(1)
      GO TO 816
C
  811 CONTINUE
      DO 815 J=1,ND
      DISP(J)=0.
      I=LM(J,N)
      IF (I.GT.0) DISP(J)=X(I)
  815 CONTINUE
C
      CALL LENTH1 (XYZ(1,N),RL)
C
      DO 814 L=1,NINT
      EP=EPSIN(N)
      R=XG(L,NINT)
      CALL DERIQ1(XYZ(1,N),R,BQ,XJ)
      DO 813 K=1,ND
  813 EP = EP + BQ(K)*DISP(K)
      STN(L)=EP
      STS(L)=E(MTYPE)*EP
      PP(L)=STS(L)*AREA(MTYPE)
      IF (IPRI.EQ.0) WRITE(6,2510) N,L,PP(L),STS(L),STN(L)
  814 CONTINUE
      IF (IPRI.EQ.0) WRITE(6,2520)
C
C
C***  DATA PORTHOLE  ***************************** (START)
C
  816 IF (JNPORT.EQ.0 .OR. KELDTE.NE.3) GO TO 830
      RECLAB=RECL84
      DO 817 L=1,NINT
      WRITE (LUI) RECLAB,L,PP(L),STS(L),STN(L)
  817 CONTINUE
C
C***  DATA PORTHOLE  ***************************** ( END )
C
C
  830 CONTINUE
C
      RETURN
C
C
C      NONLINEAR STRESS CALCULATION
C
  805 IPRNT=0
      DO 810 N=1,NUME
      IF (IDEATH.EQ.0) GO TO 910
      ETIM=DABS(ETIMV(N))
      IF (IDEATH.EQ.2) GO TO 890
      IF (TIME.LT.ETIM) GO TO 810
      GO TO 910
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
  890 IF (TIME.GT.ETIM) GO TO 810
  910 IPST=IPS(N)
      IF (IPST.EQ.0) GO TO 810
      IF(IPRI.NE.0) GO TO 803
      IPRNT=IPRNT + 1
      IF (IPRNT.NE.1) GO TO 803
      GO TO (870,870,870,875,875,875,875,870), MODEL
  870 WRITE (6,2500) NG
      GO TO 803
  875 WRITE (6,2600) NG
C
  803 IF (IPRI.EQ.0) WRITE(6,2520)
      MTYPE=MATP(N)
      IELD=IELTD(N)
      ND=IELD*3
C
      IF (ITYPT.NE.1) GO TO 850
      ND=1
      RST(1)=XYZ(1,N)
      DISP(1)=0.
      I=LM(1,N)
      IF (I.GT.0) DISP(1)=X(I)
      IF (IDEATH.NE.1) GO TO 860
      RST(1)=RST(1) + EDISB(1,N)
      DISP(1)=DISP(1) - EDISB(1,N)
      GO TO 860
C
  850 DO 851 L=1,ND
      RST(L)=XYZ(L,N)
      DISP(L)=0.
      I=LM(L,N)
      IF (I.GT.0) DISP(L)=X(I)
      IF (IDEATH.NE.1) GO TO 851
      RST(L)=RST(L) + EDISB(L,N)
      DISP(L)=DISP(L) - EDISB(L,N)
  851 CONTINUE
C
C
  860 CALL STIFN1 (N,AREA(MTYPE),PROP(1,MTYPE),WA(1,N),
     1                 NODGL(1,N),TEMPV1,TEMPV2,EPSIN(N),S,RE,0)
C
C
C***  DATA PORTHOLE  ****************************  (START)
C
      IF (JNPORT.EQ.0 .OR. KPLOTE.NE.0) GO TO 810
      RECLAB=RECLE4
      DO 880 L=1,NINT
      WRITE (LUL) RECLAB,L,PP(L),STS(L),STN(L)
  880 CONTINUE
C
C***  DATA PORTHOLE  ****************************  ( END )
C
C
  810 CONTINUE
C
      RETURN
C
  900 STOP
```

```
C
C
 1000 FORMAT (I5,4F10.0)
 1010 FORMAT (8F10.0)
 1100 FORMAT (5I5,2F10.0)
 1200 FORMAT (16I5)
C
 2000 FORMAT (////37H M A T E R I A L   C O N S T A N T S)
 2010 FORMAT (///5H  SET,10X,5H AREA,10X,5H  DEN,10X,5H     E/)
 2011 FORMAT (I5,8E15.6)
 2020 FORMAT (///5H  SET,10X,5H AREA,10X,5H  DEN,12X,5HPOINT,9X,
     1        6HSTRAIN,9X,6HSTRESS,11X,4HETAN/ I5,2E15.6,12X,I5,2E15.6)
 2021 FORMAT (47X,I5,3E15.6)
 2030 FORMAT (///27H MATERIAL CONSTANTS SET NO.,I5/
     1        14H     AREA     =,E15.6/14H     DENSITY  =,E15.6//)
 2031 FORMAT (37H     TEMP ........( PROP( 1- 6) ).. =,6(E15.6)/
     1        37H     E ...........( PROP( 7-12) ).. =,6(E15.6)/
     2        37H     ALPHA .......( PROP(13-18) ).. =,6(E15.6)/
     3        37H     REF TEMP ....( PROP(   19) ).. =,E15.6)
 2040 FORMAT (///5H  SET,10X,5H AREA,10X,5H  DEN,10X,5H    E,
     1        10X,5HYIELD,10X,5H   ET /)
 2061 FORMAT (39H     TEMP ..........( PROP( 1- 6) ).. =,6E15.6/
     1        39H     E .............( PROP( 7-12) ).. =,6E15.6/
     2        39H     YIELD .........( PROP(13-18) ).. =,6E15.6/
     3        39H     ET ............( PROP(19-24) ).. =,6E15.6/
     4        39H     ALPHA .........( PROP(25-30) ).. =,6E15.6/
     5        39H     REF TEMP ......( PROP(   31) ).. =,E15.6/
     6        39H     CREEP LAW KEY .( PROP(   32) ).. =,F8.0/
     7        39H     CREEP LAW COEFFICIENTS ............/
     8        39H     A0 ............( PROP(   33) ).. =,E15.6/
     9        39H     A1 ............( PROP(   34) ).. =,E15.6/
     A        39H     A2 ............( PROP(   35) ).. =,E15.6/
     1        39H     A3 ............( PROP(   36) ).. =,E15.6/
     2        39H     A4 ............( PROP(   37) ).. =,E15.6/
     3        39H     A5 ............( PROP(   38) ).. =,E15.6/
     4        39H     A6 ............( PROP(   39) ).. =,E15.6)
 2071 FORMAT (6H PROP(,I2,1H,I2,3H) =,E15.6)
 2200 FORMAT (///38H E L E M E N T   I N F O R M A T I O N//
     1        39X,7HINITIAL/
     2        4X,1HN,3X,4HIELD,3X,3HIPS,2X,4HMTYP,4X,2HKG,
     3        9X,6HSTRAIN,12X,A6,9X,5HETIME,13X,
     4        34HNODE(1)  NODE(2)  NODE(3)  NODE(4)/)
 2201 FORMAT (///,'E L E M E N T   N O D A L   I N F O R M A T I O N',//
     1        4X,'N',3X,'IELD',4X,'KG',9X,'ETIME',13X,'ROTOR (NODE(1))',
     2        3X,'STATOR (NODE(2))',/)
 2210 FORMAT (I5,3I6,I7,5X,2E15.6,3X,E15.6,2X,4I7)
 2211 FORMAT (I5,I6,I7,9X,E15.6,8X,I5,9X,I5)
 2300 FORMAT (///16H ELEMENT GROUP =,I2,16H  (TRUSS / RUSS)/
     1        17H ELEMENT NUMBER =,I4/
     2        7H IELD =,I3,26H IS GREATER THAN NPAR(7) =,I3/
     3        5H STOP)
 2310 FORMAT (///16H ELEMENT GROUP =,I2,16H  (TRUSS / RUSS)/
     1        17H ELEMENT NUMBER =,I4/
     2        7H MTYP =,I3,27H IS GREATER THAN NPAR(16) =,I3/5H STOP)
 2500 FORMAT (1H1,46HS T R E S S   C A L C U L A T I O N S   F O R
     1        27H E L E M E N T   G R O U P ,I5,15H  ( TRUSSES )  //,4X,
     2        7HELEMENT,10H  LOCATION,8X,5HFORCE,12X,6HSTRESS,11X,
     3        6HSTRAIN,9X,11HTEMPERATURE,/)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
                    RUSS                     DATE = 81245            18/23/01

2510 FORMAT (I9,I10,3(2X,E15.6))
2520 FORMAT (1H )
2600 FORMAT (1H1,48HS T R E S S   C A L C U L A T I O N S   F O R    ,
    1          27HE L E M E N T   G R O U P  ,I5,15H  ( TRUSSES ) ,//,4X,
    2          7HELEMENT,10H  LOCATION,5X,5HSTATE,8X,5HFORCE,12X,
    3          6HSTRESS,11X,6HSTRAIN,9X,11HTEMPERATURE,/)
C
      END
```

```
                                        RUSS                  DATE = 81245

*OPTIONS IN EFFECT*  NOTERM,ID,EBCDIC,SOURCE,NOLIST,NODECK,LOAD,NOMAP,NOTEST
*OPTIONS IN EFFECT*  NAME = RUSS     , LINECNT =        60
*STATISTICS*     SOURCE STATEMENTS =        492,PROGRAM SIZE =      21754
*STATISTICS*  NO DIAGNOSTICS GENERATED
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
C
C*************************************************************************
C*************************************************************************
C*************************************************************************
C*************************************************************************
C
      SUBROUTINE BEARNG(BD,BL,BC,VISC,PVAP,
C
     1                  TH1,TH2,PB1,PB2,NGRID,NSCLN,NPORT,
C
     2                  NFILM,KOFK,KOFC,R,IELTD,X,Y,Z,XYZ,LM,
C
     3                  S,RE,MADR,LSTIFB,NDM,NUME)
C
C.......................................................................
C.                                                                     .
C.    P R O G R A M     F U N C T I O N                                .
C.                                                                     .
C.          . TO READ BEARING ELEMENTS INFORMATION                     .
C.                                                                     .
C.          . TO CALCULATE AND ASSEMBLE ELEMENT STIFFNESSES            .
C.                                                                     .
C.      E X E C U T I O N    M O D E                                   .
C.                                                                     .
C.              IND=0      BEARING INFORMATION IS READ                 .
C.                                                                     .
C.              IND=4      ELEMENT STIFFNESSES ARE CALCULATED          .
C.                                                                     .
C.                         AND ASSEMBLED                               .
C.                                                                     .
C.......................................................................
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /EL/IND,ICOUNT,NPAR(20),NUMEG,NEGL,NEGNL,IMASS,
C
     1           IDAMP,ISTAT,NDOFDM,KLIN,IEIG,IMASSN,IDAMPN
C
      COMMON /DIM/N0,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,
C
     1           N14,N15
C
      COMMON /CONST/ DT,DTA,A0,A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,
C
     1           A11,A12,A13,A14,A15,A16,A17,A18,A19,A20,IOPE
C
      COMMON /VAR/NG,MODEX,IUPDT,KSTEP,ITEMAX,IEQREF,ITE,
C
     1           KPRI,IREF,IEQUIT,IPRI,KPLOTN,KPLOTE
C
      COMMON /FORCEB/ F1P(100),F2P(100),FR(100)
C
      COMMON /BEAR/ IBEAR,MTOTB
C
      COMMON /BEAR2/ B(1)
C
      COMMON A(1)
C
      REAL A
```

```
C
      REAL 8
C
      DIMENSION BD(1),BL(1),BC(1), VISC(1),PVAP(1),TH1(1),
C
     1          TH2(1),PB1(1),PB2(1),NGRID(1),NSOLN(1),R(1),
C
     2          NPORT(1),NFILM(1),KOFK(1),KOFC(1),IELTD(1),
C
     3          X(1),Y(1),Z(1),LM(NDM,1),XYZ(NDM,1),S(1),
C
     1          RE(1),VELB(12),ACCB(12)
C
      IF (IND.GT.0) GO TO 420
C
C.....................................................................
C.                                                                   .
C.    R E A D   A N D   G E N E R A T E   E L E M E N T              .
C.                                                                   .
C.                      I N F O R M A T I O N                        .
C.....................................................................
      WRITE (6,1999)
      N = 1
      IREAD = 5
  160 READ (IREAD,1000) M,BDE,BLE,BCE,VISCE,PVAPE,TH1E,
C
     2                  TH2E,PB1E,PB2E,NGRIDE,NSOLNE,NPORTE,
C
     3                  NFILME,KOFKE,KOFCE
C
C     GENERATE MISSING ELEMENT INFORMATION IF NECESSARY
C
  120 IF (M.NE.N) GO TO 200
      BDG       = BDE
      BLG       = BLE
      BCG       = BCE
      VISCG     = VISCE
      PVAPG     = PVAPE
      TH1G      = TH1E
      TH2G      = TH2E
      PB1G      = PB1E
      PB2G      = PB2E
      NGRIDG    = NGRIDE
      NSOLNG    = NSOLNE
      NPORTG    = NPORTE
      NFILMG    = NFILME
      KOFKG     = KOFKE
      KOFCG     = KOFCE
C
  200 BD(N)     = BDG
C
      BL(N)     = BLG
      BC(N)     = BCG
      VISC(N)   = VISCG
      PVAP(N)   = PVAPG
      TH1(N)    = TH1G
      TH2(N)    = TH2G
      PB1(N)    = PB1G
```

```
      PB2(N)    = PB2G
      NGRID(N)  = NGRIDG
      NSOLN(N)  = NSOLNG
      NPORT(N)  = NPORTG
      NFILM(N)  = NFILMG
      KOFK(N)   = KOFKG
      KOFC(N)   = KOFCG
C
      WRITE (6,2000) N,BO(N),BL(N),BC(N),VISC(N),PVAP(N),TH1(N),
C    1                TH2(N),PB1(N),PB2(N),NGRID(N),NSOLN(N),NPORT(N),
C    1                NFILM(N),KOFK(N),KOFC(N)
C                        FUTURE
C       ADD RESTART CAPABILITIES IN THE FUTURE IF NECESSARY
C
C     WRITE(6,1312) IND,N,NUME
 1312 FORMAT(5X,'IND,N,NUME=',3I10)
      IF (N.EQ.NUME) RETURN
      N = N + 1
      IF (N.GT.M) GO TO 160
      GO TO 120
C
C.........................................................................
C.                                                                       .
C.     ASSEMBLE INSTANTANEOUS FORCE VECTOR,                              .
C.              TANGENT STIFFNESS(AK) AND TANGENT DAMPING (AC)           .
C.                MATRICES OF THE BEARING ELEMENT                        .
C.                                                                       .
C.........................................................................
C
  420 IF ( IND.NE.4) RETURN
C
      ND=12
C
      DO 740 N=1,NUME
C
C     CHECK IF ELEMENT BELONGS TO THE CURRENT BLOCK
C
      CALL ECHECK (LM(1,N),ND,ICODE,IUPDT)
C
      IF (ICODE.EQ.1) IELCPL = IELCPL + 1
      IF (ICODE.EQ.1) GO TO 740
C
C                  *********FUTURE*********
C
C     ADD BIRTH AND DEATH OPTIONS LATER IF NECESSARY
C
C                  **********FUTURE********
C
C     CALCULATE INERTIAL COORDINATES OF DAMPER
C
C     INSIDE AND OUTSIDE SURFACE CENTER-LINE
C
      XHB = XYZ(1,N)
```

```
      IF (LM(1,N).GT.0) XRB = X(LM(1,N))
C     YRB = XYZ(2,N)
      IF (LM(2,N).GT.0) YRB = X(LM(2,N))
C     ZRB=XYZ(3,N)
      IF (LM(3,N).GT.0) ZRB = X(LM(3,N))
C     XSB=XYZ(4,N)
      IF (LM(7,N).GT.0) XSB = X(LM(7,N))
C     YSB=XYZ(5,N)
      IF (LM(8,N).GT.0) YSB = X(LM(8,N))
C     ZSB=XYZ(6,N)
      IF (LM(9,N).GT.0) ZSB = X(LM(9,N))
      DO 725 L=1,ND
      VELB(L)=0.
      ACCB(L)=0.
      I=LM(L,N)
      IF (I.GT.0) VELB(L)=Y(I)
      IF (I.GT.0) ACCB(L)=Z(I)
C
C
  725 CONTINUE
C
C
      XDTR  = VELB(1)
      YDTR  = VELB(2)
      ZDTR  = VELB(3)
      XDTS  = VELB(7)
      YDTS  = VELB(8)
      ZDTS  = VELB(9)
      XDT2R = ACCB(1)
      YDT2R = ACCB(2)
      ZDT2R = ACCB(3)
      XDT2S = ACCB(7)
      YDT2S = ACCB(8)
      ZDT2S = ACCB(9)
C     WRITE (6,2500) XRB,YRB,ZRB,XSB,YSB,ZSB
 2500 FORMAT (7X,'X=ROTOR',5X,'Y=ROTOR',5X,'Z=ROTOR',4X,'X=STATOR',
     1       5X,'Y=STATOR',4X,'Z=STATOR',//,5X,6(2X,D13.6))
C     WRITE (6,2550) XDTR,YDTR,ZDTR,XDTS,YDTS,ZDTS
 2550 FORMAT (7X,'ROTOR XDT',4X,'ROTOR YDT',4X,'ROTOR ZDT',
     1       3X,'STATOR XDT',4X,'STATOR YDT',3X,'STATOR ZDT',//,
     2       5X,6(2X,D13.6))
C     WRITE (6,2600) XDT2R,YDT2R,ZDT2R,XDT2S,YDT2S,ZDT2S
 2600 FORMAT (7X,'ROTOR XDT2',4X,'ROTOR YDT2',3X,'ROTOR ZDT2',
     1       3X,'STATOR SDT2',3X,'STATOR YDT2',3X,'STATOR ZDT2',//,
     2       5X,6(2X,D13.6))
C
C     GET BACK THE OTHER ELEMENT PROPERTIES TO BE
C
C     USED IN SQUEEZE
C
      BDE    = BD(N)
      BLE    = BL(N)
      BCE    = BC(N)
      VISCE  = VISC(N)
      PVAPE  = PVAP(N)
      TH1E   = TH1(N)
      TH2E   = TH2(N)
      PB1E   = PB1(N)
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
      PB2E     = PB2(N)
      NGRIDE   = NGRID(N)
      NSOLNE   = NSOLN(N)
      NPORTE   = NPORT(N)
      NFILME   = NFILM(N)
      KOFKE    = KOFK(N)
      KOFCE    = KOFC(N)
C
C
      DO 726 I=1,ND
  726 RE(I)=0.
      DO 727 I=1,78
  727 S(I)=0.
C
C   CALL SQUEEZE TO CALCULATE INSTANTANEOUS FORCE VECTOR,
C
C   STIFFNESS MATRIX(AK) AND DAMPING MATRIX (AC).
C
      CALL SQUEEZ (BDE,BLE,BCE,VISCE,TH1E,TH2E,PB1E,PB2E,
C
     1             YRB,ZRB,YDTR,ZDTR,YSB,ZSB,YDTS,ZDTS,AK11,AK12,
C
     2             AK22,AC11,AC22,F1,F2,NGRIDE,NSOLNE,NPORTE,
C
     3             KOFKE,KOFCE,NFILME,PVAPE)
C
C
C
      S(13)  = AK11
      S(14)  = AK12
      S(19)  =-AK11
      S(20)  =-AK12
      S(24)  = AK22
      S(29) =-AK12
      S(30) =-AK22
      S(64) = AK11
      S(65) = AK12
      S(69) = AK22
C
C
C
      RE(2) = -F1
      RE(3) = -F2
      RE(8) = F1
      RE(9) = F2
C
      F1P(N)=F1
      F2P(N)=F2
      FR(N)=DSQRT(F1*F1+F2*F2)
C
C     ADD ELEMENT VECTOR RE TO EXTERNAL LOAD VECTOR
C
C
C     WRITE (6,981) (S(I),I=1,78)
  981 FORMAT (5X,'BEARING STIFFNESS (BEARING):',//,
     1       8(3X,D13.6))
      CALL ADDBAN (A(MADR),A(NI),S,RE,LM(1,N),ND,2).
C
C     ADD ELEMENT STIFFNESS (AK)
```

```
C
C     WRITE (6,461) (LM(I,N),I=1,ND)
  461 FORMAT (5X,'LMARRAY(BEARING):',//,5X,8(2X,I12))
C     WRITE (6,750) IREF
  750 FORMAT (///,'$$$$$$$$$$$$$$$$$$ IREF (BEARING) $$$$$ =',I5,///)
      IF ( IREF) 641,739,641
  739 CALL ADDBAN (A(N4),A(N1),S,RE,LM(1,N),ND,1)
C
C     ADD EFFECT OF BEARING DAMPING MATRIX (AC)
C
  641 IF (KOFCE.EQ.0) GO TO 740
      DO 741 I=1,ND
  741 RE(I)=0.
      DO 742 I=1,78
  742 S(I)=0.
      RE(2)= (A4-1)*YDTR+A5*YDT2R
      RE(3)= (A4-1)*ZDTR+A5*ZDT2R
      RE(8)= (A4-1)*YDTS+A5*YDT2S
      RE(9)= (A4-1)*ZDTS+A5*ZDT2S
C
      WRITE (6,749) A4,A5,YDTR,YDT2R,YDTS,YDT2S
  749 FORMAT (5X,'A4,A5,YDTR,YDT2R,YDTS,YDT2S :',//,8(3X,D12.5))
      WRITE (6,759) (RE(I),I=1,ND)
  759 FORMAT (5X,'NUMERICAL TIME INTEGRATION VALUES OF VELOCITIES:',
     1       //,8(3X,D12.5))
      WRITE (6,751) AC11,AC22
  751 FORMAT (5X,'DAMPING MATRIX :',2(10X,D13.6))
      S(13)=AC11
      S(19)=-AC11
      S(24)=AC22
      S(30)=-AC22
      S(64)=AC11
      S(69)=AC22
C
C
      LL=0
      LJ=1
      DO 743 L=1,ND
      I=LM(L,N)
      DO 744 K=L,ND
      LL=LL+1
      IF (I.EQ.0) GO TO 744
      R(I)=R(I)-S(LL)*RE(K)
C     WRITE (6,752) L,I,K,LL,R(I),S(LL),RE(K)
  752 FORMAT (5X,'L,I,K,LL,R(I),S(LL),RE(K) :',
     1       //,35X,4(4X,I5),10X,3(4X,D12.5))
  744 CONTINUE
      IF (L.EQ.ND) GO TO 743
      L1=L+1
      DO 745 J=L1,ND
      LJ=LJ+1
      IF (J.EQ.ND) LJ=LJ+1
      IJ=LM(J,N)
      IF (IJ.EQ.0) GO TO 745
      R(IJ)=R(IJ)-S(LJ)*RE(L)
C     WRITE (6,765) L,L1,J,LJ,IJ,R(IJ),S(LJ),RE(L)
  765 FORMAT (5X,'L,L1,J,LJ,IJ,R(IJ),S(LJ),RE(L) :',
     1       //,35X,5(4X,I5),3(4X,D12.5))
```

```
  745 CONTINUE
  743 CONTINUE
      WRITE (6,753) (R(I),I=1,4)
  753 FORMAT (5X,'DAMPED EXTERNAL LOAD VECTOR',//,8(3X,D12.5))
  740 CONTINUE
C
      RETURN
C
 1000 FORMAT (I10,6F10.0,/,3F10.0,6I5)
C
 1999 FORMAT (//////,4X,'BEARING ELEMENT INFORMATION',//,4X,
C
     1            'N     BD          BL          BC          VISC       ',
C
     2       ,'PVAP        TH1         TH2         PB1         PB2         NG
C
     3  NS     NP     NF   KK    KC ')
C
 2000 FORMAT (I5,9D11.4,4I5,2I4)
C
      END
```

```
FORTRAN IV G1  RELEASE 2.0                BEARNG               DATE = 81229

  *OPTIONS IN EFFECT*  NOTERM,ID,EBCDIC,SOURCE,NOLIST,NODECK,LOAD,NOMAP
  *OPTIONS IN EFFECT*  NAME = BEARNG  , LINECNT =       60
  *STATISTICS*     SOURCE STATEMENTS =       182,PROGRAM SIZE =      6808
  *STATISTICS*  NO DIAGNOSTICS GENERATED
```

```
C
C
C
C**********************************************************************
C**********************************************************************
C**********************************************************************
C
      SUBROUTINE SIZEB(N)
C
      IMPLICIT REAL *8(A-H,O-Z)
C
      COMMON B(1)
C
      COMMON /BEAR/ IBEAR, MTOTB
C
      WRITE (6,1000)
      WRITE (6,1001) N
      IF (N.LT.MTOTB) GO TO 20
      WRITE (6,1004) N
      STOP
   20 WRITE (6,1008)
 1000 FORMAT (/,4X,'CORE INFORMATION        (DECIMAL)...')
C
 1001 FORMAT (4X,'     REQUESTED CORE FOR BEARING INFORMATION =',5X,I8)
C
 1004 FORMAT (4X,'     NOT AVAILABLE   STOP',/,4X,'****CHANGE MTOTB',
C
     1          '(IN MAIN) TO (AT LEAST)    ',I8,'*****')
C
 1008 FORMAT (4X,'     OBTAINED        ',//)
      RETURN
      END
```

```
FORTRAN IV G1   RELEASE 2.0                SIZEB                    DATE = 81229

   *OPTIONS IN EFFECT*  NOTERM,ID,EBCDIC,SOURCE,NOLIST,NODECK,LOAD,NOMAP
   *OPTIONS IN EFFECT*  NAME = SIZEB    , LINECNT =        60
   *STATISTICS*    SOURCE STATEMENTS =        16,PROGRAM SIZE =       640
   *STATISTICS*  NO DIAGNOSTICS GENERATED
```

```
C
C*****************************************************************
C*****************************************************************
C*****************************************************************
C
      SUBROUTINE SQUEEZ(AD,AL,AC,AISC,ATH1,ATH2,AB1,AB2,U,V,
C    1 UDT,VDT,UB,VB,UBT,VBT,AK11,AK12,AK22,AC11,AC22,F1,F2,NGRIA,NSO_A,
C    2 NPORA,KAFK,KAFC,NFILA,PVAA)
C
C     NONLINEAR TIME-TRANSIENT SQUEEZE-FILM DAMPER INTERACTIVE ELEMENT
C
C     THIS CODE COMPUTES INSTANTANEOUS FORCE VECTOR AND ITS
C
C     SPATIAL GRADIENTS, I.E., THE TANGENT STIFFNESS AND DAMPING
C
C     MATRICES.
C
C
C     -------------
C     -------------
C     NOMENCLATURE
C     -------------
C     -------------
C
C     INPUT
C     -----
C
C     BD=NOMINAL DAMPER ANNULUS DIAMETER(IN)
C
C     BL=NOMINAL DAMPER ANNULUS LENGTH(IN)
C
C     BC=DAMPER ANNULUS RADIAL CLEARANCE(IN)
C
C     VISC=DAMPER LUBRICANT VISCOSITY(REYNS)
C
C     PVAP=FILM RUPTURE PRESSURE(PSIA)
C
C     THT(1)=POSITION ANGLE OF LUBRICANT PORT-1(DEG)
C
C     THT(2)=POSITION ANGLE OF LUBRICANT PORT-2(DEG)
C
C     PB(1)=SPECIFIED BOUNDARY PRESSURE AT PORT-1(PSIA)
C
C     PB(2)=SPECIFIED BOUNDARY PRESSURE AT PORT-2(PSIA)
C
C     NGRID=NUMBER OF FINITE-DIFFERENCE GRID POINTS PER DAMPER ARC(ODD)
C
C     NSOLN=1,LONG-BEARING SOLUTION USED
C
C          =2,SHORT-BEARING(PARABOLIC) SOLUTION USED
C
C          =3,FOURIER-SERIES 2-D CONVERGENT SOLUTION USED
C
C     NPORT=NUMBER OF LUBRICANT PORTS(0,1 OR 2)
C
C          IF NPORT=0, JOINED-BOUNDARY CONDITION IS USED
```

SQUEEZ DATE = 81229 09/18/44

```
C     NFILM=NUMBER OF IDENTICAL ANNULI FOR THE DAMPER
C     KOFK=0,STIFFNESS MATRIX NOT COMPUTED
C     KOFK=1,STIFFNESS MATRIX COMPUTED
C     KOFC=0,DAMPING MATRIX NOT COMPUTED
C     KOFC=1,DAMPING MATRIX COMPUTED
C     X=X-INERTIAL COORDINATE OF DAMPER INSIDE SURFACE CENTER-LINE(IN)
C     Y=Y-INERTIAL COORDINATE OF DAMPER INSIDE SURFACE CENTER-LINE(IN)
C     XDT=X-INERTIAL VELOCITY OF INSIDE SURFACE CENTER-LINE(IN/SEC)
C     YDT=Y-INERTIAL VELOCITY OF INSIDE SURFACE CENTER-LINE(IN/SEC)
C     XB=X-INERTIAL COORDINATE OF DAMPER OUTSIDE SURFACE CENTER-LINE(IN)
C     YB=Y-INERTIAL COORDINATE OF DAMPER OUTSIDE SURFACE CENTER-LINE(IN)
C     XBT=X-INERTIAL VELOCITY OF DAMPER OUTSIDE SURFACE CENTER-LINE(IN)
C     YBT=Y-INERTIAL VELOCITY OF    MPER OUTSIDE SURFACE CENTER-LINE(IN)
C     OUTPUT
C     ------
C     F1=X-FORCE COMPONENT ON INSIDE DAMPER SURFACE(LBS)
C     F2=Y-FORCE COMPONENT ON INSIDE DAMPER SURFACE(LBS)
C     FORCE COMPONENTS ON OUTSIDE DAMPER SURFACE ARE EQUAL BUT OPPOSITE
C     F1 AND F2 BECAUSE FLUID INERTIA EFFECTS ARE NEGLECTED
C     SYMMETRIC PORTION OF STIFFNESS MATRIX:
C          AK11=KXX(LBS/IN)
C          AK12=KXY(LBS/IN)=KYX
C          AK22=KYY(LBS/IN)
C     DIAGONAL PORTION OF DAMPING MATRIX
C          AC11=CXX(LB*SEC/IN)
C          AC22=CYY(LB*SEC/IN)
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON/INPUT1/BD,BL,BC,VISC,THT(2),PB(2),PVAP
C
      COMMON/INPUT2/NGRID,NSOLN,NPORT,NFILM
C
```

```
      COMMON/COORD/X,Y,XDT,YDT,XB,YB,XBT,YBT
C     COMMON/FILM/TH(101),H(101),DHDX(101),DHDT(101),STH(101),CTH(101),
C    1DXD(2),ALFA(2)
C     COMMON/WORK/FNGDM1,KB,KOUNT
C     COMMON/INC/HMIN,VEL,DELS,DELST
C     DIMENSION A(101),B(101),C(101),E(101),RH(101),P(101),ARG1(101),
C    1 ARG2(101),D(101)
C
C     ALLOCATE INPUT NAMES
C
      BD     = AD
      BL     = AL
      BC     = AC
      VISC   = AISC
      THT(1)= ATH1
      THT(2)= ATH2
      PB(1)  = AB1
      PB(2)  = AB2
      PVAP   = PVAA
      X      = U
      Y      = V
      XDT    = UDT
      YDT    = VDT
      XB     = UB
      YB     = VB
      XBT    = UBT
      YBT    = VBT
      NGRID = NGRIA
      NSOLN = NSOLA
      NPORT = NPORA
      KOFK   = KAFK
      KOFC   = KAFC
      NFILM = NFILA
      PI     = 3.141592654
      WRITE (6,2500) X,Y,XB,YB,XDT,YDT,XBT,YBT
C2500 FORMAT (7X,'X-ROTOR',5X,'Y-ROTOR',5X,'X-STATOR',4X,'Y-STATOR',
     3         5X,'ROTOR XDT',
     1         5X,'ROTOR YDT',4X,'STATOR XDT',5X,'STATOR YDT',//,
     2         5X,8(2X,D13.6))
C
C     WRITE (6,1999)
C1999 FORMAT (//////,4X,'BEARING ELEMENT INFORMATION',//,4X,
C    1              ' BD          BL           BC            VISC     '
C    2              ,'PVAP        TH1          TH2           PB1            PB2
C    3  NS    NP    NF    KK    KC')
C     WRITE(6,10)BD,BL,BC,VISC,PVAP,THT(1),THT(2),PB(1),PB(2),
C    1           NGRID,NSOLN,NPORT,NFILM,KOFK,KOFC
   10 FORMAT (9D11.4,4I5,2I4)
C
```

```
C       SET UP
C
        KOUNT   = 1
        MTEST   = NGRID + 1
        NTEST   = MTEST/2
        KTEST   = 2*NTEST
C
        IF(KTEST.NE.MTEST) NGRID=NGRID-1
C
        FNGDM1  = FLOAT(NGRID-1)
        THT(1)  = THT(1)*PI/180.
C
        IF(NPORT.LT.2) GO TO 20
C
        THT(2)  = THT(2)*PI/180.
C
        IF(THT(2).LT.THT(1)) THT(2)=THT(2)+2.*PI
C
        ALFA(1) = THT(2)-THT(1)
        ALFA(2) = 2.*PI-ALFA(1)
        DXD(1)  = 0.5*BD*ALFA(1)/FNGDM1
        DXD(2)  = 0.5*BD*ALFA(2)/FNGDM1
C
        GO TO 40
C
   20   DXD(1)  = BD*PI/FNGDM1
        ALFA(1) = 2.*PI
C
   40   CONTINUE
C
C
        WRITE(6,10)DXD(1),DXD(2)
        AKXX = 0.0
        AKXY = 0.0
        AKYX = 0.0
        AKYY = 0.0
        ACXX = 0.0
        ACXY = 0.0
        ACYX = 0.0
        ACYY = 0.0
        FX   = 0.0
        FY   = 0.0
C
C       BRANCH ACCORDING TO SOLUTION DESIGNATED
C
   60   CONTINUE
        CALL INCRNT
        GO TO (100,100,300),NSOLN
  100   CONTINUE
C
C       SOLVE FOR SQUEEZE FILM PRESSURE DISTRIBUTION
C
        DO 190 KB=1,NPORT
        P(1)=PB(KB)
        IF(KB.EQ.1) P(NGRID)=PB(2)
        IF(KB.EQ.2) P(NGRID)=PB(1)
        DX=DXD(KB)
        CALL DFILM
        A(2)=0.0
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
      B(2)   = P(1)
      DO 110 K=2,NGRID
      COEF1 = H(K)**3/DX**2
      COEF2 = (1.5*H(K)**2)*DHDX(K)/DX
      C(K)  =-2.*COEF1
      D(K)  = COEF1+COEF2
      E(K)  = COEF1-COEF2
      RH(K) = 12.*VISC*DHDT(K)
      IF(NSOLN.EQ.1) GO TO 110
      C(K)  = C(K)-(8.*H(K)**3)/BL**2
  110 CONTINUE
      NGRD  = NGRID-1
      DO 120 K=2,NGRD
      FOCTR = C(K)+E(K)*A(K)
      A(K+1)=-D(K)/FOCTR
  120 B(K+1)=(RH(K)-E(K)*B(K))/FOCTR
      NGR   = NGRID-2
      DO 130 K=1,NGR
      J     = NGRID-K
      P(J)  = A(J+1)*P(J+1)+B(J+1)
      IF(P(J).LT.PVAP) P(J)=PVAP
  130 CONTINUE
C     WRITE(6,135)(P(J),J=1,NGRID)
  135 FORMAT(2X,14E9.2)
C
C     INTEGRATE PRESSURE DISTRIBUTION TO GET X AND ' FORCE COMPONENTS
C
      DO 140 K=1,NGRID
      ARG1(K)   = P(K)*CTH(K)
  140 ARG2(K)   = P(K)*STH(K)
      A1        = ARG1(1)+ARG1(NGRID)
      A2        = ARG2(1)+ARG2(NGRID)
      B1        = 0.0
      B2        = 0.0
      DO 150 K=2,NGRD,2
      B2 = B2+ARG2(K)
  150 B1 = B1+ARG1(K)
      C1 = 0.0
      C2 = 0.0
      DO 160 K=3,NGR,2
      C1=C1+ARG1(K)
  160 C2=C2+ARG2(K)
      DTHET=2.*DX/BD
      FACTR=FLOAT(NFILM)*DTHET/3.
      XQ=FACTR*(A1+4.*B1+2.*C1)
      YQ=FACTR*(A2+4.*B2+2.*C2)
      IF(NSOLN.EQ.1) FACTR=-BD*BL/2.
      IF(NSOLN.EQ.2) FACTR=-BD*BL/3.
      XQ=XQ*FACTR
      YQ=YQ*FACTR
      FX=FX+XQ
      FY=FY+YQ
  190 CONTINUE
      GO TO (500,520,540,560,580),KOUNT
  500 F1=FX
      F2=FY
      IF((KOFK.EQ.0).AND.(KOFC.EQ.0)) RETURN
      IF(KOFK.EQ.0) KOUNT=4
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
      IF(KOFK.EQ.0) GO TO 60
      GO TO 590
  520 AKXX=(FX-F1)/DELS
      AKYX=(FY-F2)/DELS
      GO TO 590
  540 AKXY=(FX-F1)/DELS
      AKYY=(FY-F2)/DELS
C     WRITE (6,657) AKXX,AKXY,AKYY
  657 FORMAT (5X,'BEARING STIFF.(SQUEEZ):   ',3(5X,D13.6))
      AK11=-AKXX
      AK22=-AKYY
      AK12=-0.5*(AKXY+AKYX)
C     WRITE (6,657) AK11,AK12,AK22
C     WRITE (6,658) F1,F2
  658 FORMAT (5X,'FORCES ON THE ROTOR:',//,8(2X,D13.5))
      IF(KOFC.EQ.0) RETURN
      GO TO 590
  560 ACXX=(FX-F1)/DELST
      ACYX=(FY-F2)/DELST
      GO TO 590
  580 ACXY=(FX-F1)/DELST
      ACYY=(FY-F2)/DELST
      GO TO 600
  590 KOUNT=KOUNT+1
      GO TO 60
  600 CONTINUE
      AC11=-ACXX
      AC22=-ACYY
      RETURN
  300 WRITE(6,700)
      RETURN
  700 FORMAT(1H1//5X'FOURIER-SERIES 2-D OPTION NOT READY FOR USE'//)
      END
```

```
FORTRAN IV G1   RELEASE 2.0          SQUEEZ              DATE = 81229

   *OPTIONS IN EFFECT*  NOTERM,ID,EBCDIC,SOURCE,NOLIST,NODECK,LOAD,NOMAP
   *OPTIONS IN EFFECT*  NAME = SQUEEZ , LINECNT =       60
   *STATISTICS*    SOURCE STATEMENTS =       159,PROGRAM SIZE =    11932
   *STATISTICS*  NO DIAGNOSTICS GENERATED
```

MAIN   DATE = 81229   09/18/44

```
C
C*****************************************************************************
C*****************************************************************************
C*****************************************************************************
C
      SUBROUTINE DFILM
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON/INPUT1/BD,BL,BC,VISC,THT(2),PB(2),PVAP
C
      COMMON/INPUT2/NGRID,NSOLN,NPORT,NFILM
C
      COMMON/COORD/X,Y,XDT,YDT,XB,YB,XBT,YBT
C
      COMMON/FILM/TH(101),H(101),DHDX(101),DHDT(101),STH(101),CTH(101),
C    1             DXD(2),ALFA(2)
C
      COMMON/WORK/FNGDM1,KB,KOUNT
C
C     COMPUTE DISTRIBUTIONS FOR H,DHDX AND DHDT
C
      R=BD/2.
      DANG=ALFA(KB)/FNGDM1
      IF(KB.EQ.1) TH(1)=THT(1)
      IF(KB.EQ.2) TH(1)=THT(2)
      DO 50 K=2,NGRID
   50 TH(K)=TH(K-1)+DANG
      CDG=DCOS(DANG)
      SDG=DSIN(DANG)
      CTH(1)=DCOS(TH(1))
      STH(1)=DSIN(TH(1))
      DO 100 K=2,NGRID
      CTH(K)=CTH(K-1)*CDG-STH(K-1)*SDG
  100 STH(K)=STH(K-1)*CDG+CTH(K-1)*SDG
      DO 150 K=1,NGRID
      H(K)=BC-(X-XB)*CTH(K)-(Y-YB)*STH(K)
      IF(H(K).LT.0.) GO TO 500
      DHDX(K)=((X-XB)*STH(K)-(Y-YB)*CTH(K))/R
  150 DHDT(K)=-(XDT-XBT)*CTH(K)-(YDT-YBT)*STH(K)
      RETURN
  500 WRITE(6,600) KB,K
      STOP
  501 RETURN
  600 FORMAT(1H1///5X'NEGATIVE FILM THICKNESS ENCOUNTERED AT KB,K=',
     1 2I10)
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
FORTRAN IV G1  RELEASE 2.0                 DEILM                     DATE = 81229

  *OPTIONS IN EFFECT*  NOTERM,ID,EBCDIC,SOURCE,NOLIST,NODECK,LOAD,NOMAP
  *OPTIONS IN EFFECT*  NAME = DEILM      , LINECNT =        60
  *STATISTICS*     SOURCE STATEMENTS =          31,PROGRAM SIZE =      1034
  *STATISTICS*  NO DIAGNOSTICS GENERATED
```

```
C
C**********************************************************************
C**********************************************************************
C**********************************************************************
C
      SUBROUTINE INCRNT
C
C     THIS SUBROUTINE INCREMENTS DISPLACEMENTS AND VELOCITIES FOR
C     THE PURPOSE OF NUMERICALLY DIFFERENTIATING THE SQUEEZE FILM
C     FORCE  VECTOR.
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON/INPUT1/BD,BL,BC,VISC,THT(2),PB(2),PVAP
C
      COMMON/COORD/X,Y,XDT,YDT,XB,YB,XBT,YBT
C
      COMMON/WORK/FNGDM1,KB,KOUNT
C
      COMMON/INC/HMIN,VEL,DELS,DELST
C
      IF(KOUNT.NE.1) GO TO 10
      SX=X
      SY=Y
      SXDT=XDT
      SYDT=YDT
      RETURN
   10 CONTINUE
      X=SX
      Y=SY
      XDT=SXDT
      YDT=SYDT
      HMIN=C-DSQRT((X-XB)**2+(Y-YB)**2)
      VEL=DSQRT((XDT-XBT)**2+(YDT-YBT)**2)
      DELS=0.05*HMIN
      DELST=0.05*VEL
      IF(KOUNT.EQ.2) X=X+DELS
      IF(KOUNT.EQ.3) Y=Y+DELS
      IF(KOUNT.EQ.4) XDT=XDT+DELST
      IF(KOUNT.EQ.5) YDT=YDT+DELST
      RETURN
      END
```

```
FORTRAN IV G1  RELEASE 2.0          INCRNT                DATE = 81229

    *OPTIONS IN EFFECT*  NOTERM,ID,EBCDIC,SOURCE,NOLIST,NODECK,LOAD,NOMAP
    *OPTIONS IN EFFECT*  NAME = INCRNT   , LINECNT =        60
    *STATISTICS*    SOURCE STATEMENTS =          27,PROGRAM SIZE =        740
    *STATISTICS*  NO DIAGNOSTICS GENERATED
```

MAIN . DATE = 81229 09/18/44

```
C           DATA SET TRUSS       AT LEVEL TMP AS OF 08/17/81
C *CDC* *DECK UVL20
C *CDC*          OVERLAY (ADINA,2,0)
C *CDC* *DECK TRUSS
C *UNI* )FOR,IS   N.TRUSS,  R.TRUSS
C *CDC*          PROGRAM TRUSS
      SUBROUTINE TRUSS
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
C
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                     .
C .   S T O R A G E                                                     .
C .                                                                     .
C .      N102    DEN     (NECESSARY FOR STATIC PROBLEMS, TOO)           .
C .      N103    AREA                                                   .
C .      N105    LM      CONNECTIVITY                                   .
C .      N106    XYZ     ELEMENT NODAL COORDINATES                      .
C .      N107    MATP    MATERIAL PROPERTY SET NUMBER                   .
C .      N108    EPSIN   INITIAL STRAIN                                 .
C .      N109    IPS     STRESS PRINTING FLAG                           .
C .      N110    ETIMV   ELEMENT BIRTH/DEATH TIME (IF IDEATH.GT.0)      .
C .      N111    EDISB   ELEMENT DISPS AT BIRTH TIME (IF IDEATH.EQ.1)   .
C .      N112    WA=IWA  WORKING ARRAY (SEE IDWAS)                      .
C .      N113    PROP    MATERIAL CONSTANTS (SEE NMCON)                 .
C .      N114    NODGL   GLOBAL NODE NUMBERS (SEE NDWS)                 .
C .      N115    IELTD   ELEMENT NUMBER OF NODES                        .
C .                                                                     .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C
C
      COMMON /DIM/ N0,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /SOL/ NUMNP,NEQ,NWK,NWM,NWC,NUMEST,MIDEST,MAXEST,NSTE,MA
      COMMON /EL/ IND,ICOUNT,NPAR(20),NUMEG,NEGL,NEGNL,IMASS,IDAMP,ISTAT
     1            ,NDOF,KLIN,IEIG,IMASSN,IDAMPN
      COMMON /VAR/ NG,MODEX,IUPDT,KSTEP,ITEMAX,IEQREF,ITE,KPRI,
     1             IREF,IEQUIT,IPRI,KPLOTN,KPLOTE
      COMMON /DPR/ ITWO
      COMMON /JUNK/ HED(12),MTOT
      COMMON /ELGLTH/ NFIRST,NLAST,NBCEL
      COMMON /ELSTP / TIME,IDTHF
      COMMON /PRCON/ IDATWR,IPRIC,NPB,IDC,IVC,IAC,IPC,IPNODE(3,15)
      COMMON /PORT/ INPORT,JNPORT,NPUTSV,LUNODE,LU1,LU2,LU3,JDC,JVC,JAC
      COMMON /TMODEL/ LINEL,NONEL,LTEL,ISPEL,KINPEL,IEPCL,IEPCK,MODMAX
      COMMON A(1)
      COMMON /TEMPRT/ TEMP1,TEMP2,ITEMPR,ITP96,N6A,N6B
      REAL A
      DIMENSION IA(1)
      EQUIVALENCE (A(1),IA(1))
C
      DIMENSION DATA(20)
C
      EQUIVALENCE (NPAR(1),NPAR1), (NPAR(2),NUME), (NPAR(3),INDNL),
     1            (NPAR(4),IDEATH), (NPAR(5),ITYPT), (NPAR(7),MXNODS),
     2            (NPAR(10),NINT), (NPAR(15),MODEL), (NPAR(16),NUMMAT),
     3            (NPAR(17),NCON), (NPAR(19),IITEMP)
C
```

```
      DIMENSION NMCON(8),IDWAS(8),NDWS(8)
      DATA RECLB1 /8HTYPE = 1/
C
      DATA NMCON / 1, 0,19, 3, 3,39,39, 0/
      DATA IDWAS / 0, 0, 0, 2, 2, 9, 9, 0/
      DATA NDWS  / 0, 0, 1, 0, 0, 1, 1, 0/
C
C
      IF (IND.NE.0) GO TO 100
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C *    I N P U T   P H A S E                                             *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C     CHECK ON RANGE AND SET DEFAULTS FOR NPAR VECTOR
C
      ISTOP=0
C
      LINEL=1
      NONEL=2
      ITEL=3
      ISPEL=4
      KINPEL=5
      IEPCI=6
      IEPCK=7
      MODMAX=8
C
      IF (NUME.GT.0) GO TO 10
      ISTOP=ISTOP+1
      IF (ISTOP.EQ.1) WRITE (6,2100) NG
      ISUB=2
      IRANGE=1
      WRITE (6,2400) ISTOP,ISUB,IRANGE,ISUB,NPAR(ISUB)
C
   10 IF (INDNL.GE.0 .AND. INDNL.LE.2) GO TO 15
      ISTOP=ISTOP+1
      IF (ISTOP.EQ.1) WRITE (6,2100) NG
      ISUB=3
      WRITE (6,2200) ISTOP,ISUB,ISUB,NPAR(ISUB)
C
   15 IF (IDEATH.NE.0) IDTHF=1
      IF (IDEATH.GE.0 .AND. IDEATH.LE.2) GO TO 20
      ISTOP=ISTOP+1
      IF (ISTOP.EQ.1) WRITE (6,2100) NG
      ISUB=4
      WRITE (6,2200) ISTOP,ISUB,ISUB,NPAR(ISUB)
C
   20 IF (ITYPT.EQ.3) GO TO 25
      IF (ITYPT.GE.0.AND.ITYPT.LE.1) GO TO 25
      ISTOP=ISTOP+1
      IF (ISTOP.EQ.1) WRITE (6,2100) NG
      ISUB=5
      WRITE (6,2200) ISTOP,ISUB,ISUB,NPAR(ISUB)
C
   25 IF (MXNODS.LE.0) MXNODS=2
      IF (MXNODS.LE.4) GO TO 30
      ISTOP=ISTOP+1
      IF (ISTOP.EQ.1) WRITE (6,2100) NG
```

```
      ISUB=7
      IRANGE=4
      WRITE (6,2300) ISTOP,ISUB,IRANGE,ISUB,NPAR(ISUB)
C
   30 NIPT=1
      IF (ITYPT.EQ.1) GO TO 34
      IF (MXNODS.GT.2) GO TO 32
      IF (MODEL.EQ.ITEL) GO TO 32
      IF (MODEL.EQ.IEPCI .OR. MODEL.EQ.IEPCK) GO TO 32
      GO TO 34
   32 NIPT=NINT
      IF (NINT.LE.0) NIPT=2
   34 NINT=NIPT
      IF (NINT.LE.4) GO TO 35
      ISTOP=ISTOP+1
      IF (ISTOP.EQ.1) WRITE (6,2100) NG
      ISUB=10
      IRANGE=4
      WRITE (6,2300) ISTOP,ISUB,IRANGE,ISUB,NPAR(ISUB)
C
   35 IF (MODEL.LE.0) MODEL=1
      IF (MODEL.LE.MODMAX) GO TO 40
      ISTOP=ISTOP+1
      IF (ISTOP.EQ.1) WRITE (6,2100) NG
      ISUB=15
      WRITE (6,2300) ISTOP,ISUB,MODMAX,ISUB,NPAR(ISUB)
C
   40 IF (NUMMAT.LE.0) NUMMAT=1
C
      IF (MODEL.EQ.MODMAX) GO TO 50
      IF (MODEL.NE.NONEL) GO TO 45
      IF (NCON.GE.4) GO TO 50
      ISTOP=ISTOP+1
      IF (ISTOP.EQ.1) WRITE (6,2100) NG
      ISUB=17
      IRANGE=4
      WRITE (6,2400) ISTOP,ISUB,NPAR(ISUB),IRANGE
      GO TO 50
C
   45 NCON=NMCON(MODEL)
C
C     CHECK ON COMPATIBILITY BETWEEN ELEMENTS OF NPAR
C
C
C        1. COMPATIBILITY OF INDNL AND IDEATH
C
   50 ISUB=3
      IF (INDNL.GT.0) GO TO 55
      IF (IDEATH.EQ.0) GO TO 52
      ISTOP=ISTOP+1
      IF (ISTOP.EQ.1) WRITE (6,2100) NG
      ISUD=4
      WRITE (6,2500) ISTOP,ISUB,NPAR(ISUB),ISUD,NPAR(ISUD)
C
C        2. COMPATIBILITY OF INDNL AND ITYPT
C
   52 IF (ITYPT.NE.2) GO TO 54
      ISTOP=ISTOP+1
```

```
      IF (ISTOP.EQ.1) WRITE (6,2100) NG
      ISUD=5
      WRITE (6,2500) ISTOP,ISUB,NPAR(ISUB),ISUD,NPAR(ISUD)
C
C        3. COMPATIBILITY OF INDNL AND MODEL
C
   54 IF (MODEL.EQ.LINEL) GO TO 55
      ISTOP=ISTOP+1
      IF (ISTOP.EQ.1) WRITE (6,2100) NG
      ISUD=L5
      WRITE (6,2500) ISTOP,ISUB,NPAR(ISUB),ISUD,NPAR(ISUD)
C
C        4. COMPATIBILITY OF MXNODS AND ITYPT
C
   55 ISUB=5
      ISUD=7
      IF (MXNODS.NE.1) GO TO 60
      IF (ITYPT.EQ.1) GO TO 60
      ISTOP=ISTOP+1
      IF (ISTOP.EQ.1) WRITE (6,2100) NG
      WRITE (6,2500) ISTOP,ISUB,NPAR(ISUB),ISUD,NPAR(ISUD)
C
C
C     CHECK FOR TEMPERATURE TAPE
C
   60 ITEMPR=0
      IF (MODEL.EQ.ITEL) ITEMPR=1
      IF (MODEL.EQ.IEPCI .OR. MODEL.EQ.IEPCK) ITEMPR=2
      IF (ITEMPR.EQ.0) GO TO 70
      IF (ITP96.GT.0) GO TO 70
      ISTOP=ISTOP+1
      IF (ISTOP.EQ.1) WRITE (6,2100) NG
      WRITE (6,2600) ISTOP
C
   70 IITEMP=ITEMPR
C
      IF (ISTOP.EQ.0) GO TO 75
      WRITE (6,2700) ISTOP
      INPUT=5
      BACKSPACE INPUT
      READ (5,1000) DATA
      WRITE (6,2800) (I,I=1,8),DATA
      IDATWR=1
C
   75 IF (IDATWR.GT.1) GO TO 90
C
C     PRINT OUT NPAR VECTOR
C
      WRITE (6,2900) NPAR1,NUME,INDNL,IDEATH
      WRITE (6,2920) ITYPT,MXNODS,NINT
      WRITE (6,2940) MODEL,NUMMAT,NCON
C
   90 IF (ISTOP.EQ.0) GO TO 95
      WRITE (6,2750)
      STOP
C
C
C***  DATA PURTHULE  *************************** (START)
```

```
C
   95 IF (JNPORT.EQ.0 .OR. NPUTSV.EQ.0) GO TO 100
      RECLAB=RECLB1
      WRITE (LU1) RECLAB,NG,(NPAR(I),I=1,20)
C
C***  DATA PORTHOLE  ***************************** ( END )
C
C
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .  E N D   O F   C H E C K   O N   N P A R   V E C T O R         .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C
C
  100 NDM=3*MXNODS
      IF (ITYPT.EQ.3) NDM=12
      IF (ITYPT.EQ.1) NDM=1
      NDW=NDWS(MODEL)
      IDWA=IDWAS(MODEL)*NINT
C
      NFIRST=N6
      IF ( IND.EQ.4) NFIRST=N10
      N101=NFIRST + 20
      N102=N101
      N103=N102 + NUMMAT*ITWO
      N105=N103 + NUMMAT*ITWO
      N106=N105 + NDM*NUME
      N107=N106 + NDM*NUME*ITWO
      N108=N107 + NUME
      N109=N108 + NUME*ITWO
      N110=N109 + NUME
      MM=0
      IF (IDEATH.GT.0) MM=1
      N111=N110 + MM*NUME*ITWO
      MM=0
      IF (IDEATH.EQ.1) MM=1
      N112=N111 + MM*NDM*NUME*ITWO
      N113=N112 + NUME*IDWA*ITWO
      N114=N113 + NGON*NUMMAT*ITWO
      N115=N114 + NDW*NUME*MXNODS
      N116=N115 + NUME
      NLAST=N116
C
C *CDC*          IF (NLAST.GT.MTOT) CALL SIZE(NLAST+2000)
      IF (IND.NE.0) GO TO 105
      MIDEST=(NLAST-NFIRST) + 1
      IF (IDATWR.LE.1) WRITE (6,2000) NG,MIDEST
      WRITE (6,2030)
      CALL SIZE (NLAST)
C
  105 IF (IND.GT.3) GO TO 110
      M2=N2
      M3=N3
      M4=N4
      GO TO 120
  110 M2=N2
      M3=N7
      M4=N8
      IF (ICOUNT.LT.3) GO TO 120
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
      M2=N6
C
  120 CALL RUSS (A(N1),A(M2),A(M3),A(M4),A(N5),A(N101),A(N102),A(N103),
     1            A(N105),A(N106),A(N107),A(N108),A(N109),A(N110),
     2            A(N111),A(N112),A(N113),A(N114),A(N115),NCON,NDOF,NDM,
     3            IDWA,A(N6A+ITWO),A(N6B+ITWO),MXNODS)
      IF (IND.GT.0) GO TO 150
      DO 140 I=1,20
  140 IA(NFIRST + I - 1)=NPAR(I)
  150 CONTINUE
C
      RETURN
C
 1000 FORMAT (20A4)
C
 2000 FORMAT (//49H LENGTH OF ARRAY NEEDED FOR STORING ELEMENT GROUP/
     3           12H DATA (GROUP,I3,26H). . . . . . . . . . . . . .
     4           15H( MIDEST ). . =,I5)
 2030 FORMAT (//49H AVAILABILITY OF CORE STORAGE FOR ELEMENT GROUP/
     1           49H DATA . . . . . . . . . . . . . . . . ( NLAST ) .)
C
 2100 FORMAT (1H1,45HERROR IN ELEMENT GROUP CONTROL CARDS (TRUSS) /
     1           16H ELEMENT GROUP =, I5/)
 2200 FORMAT (I5,7H. NPAR(,I2,27H) IS OUT OF RANGE ... NPAR(,I2,
     1           3H) =,I5)
 2300 FORMAT (I5,7H. NPAR(,I2,16H) SHOULD BE .LE.. I2,10H ... NPAR(,I2,
     1           3H) =,I5)
 2400 FORMAT (I5,7H. NPAR(,I2,16H) SHOULD BE .GE.. I2,10H ... NPAR(,I2,
     1           3H) =,I5)
 2500 FORMAT (I5,7H. NPAR(,I2,3H) =,I2,10H AND NPAR(,I2,3H) =,I2,
     1           19H ARE NOT COMPATIBLE )
 2600 FORMAT (I5,37H. TEMPERATURE TAPE SHOULD BE PROVIDED )
 2700 FORMAT (//25H TOTAL NUMBER OF ERRORS =,I5//
     1           48H CARD IMAGE LISTING AND PRINT-OUT OF NPAR VECTOR/
     2           48H (WITH DEFAULTS ENFORCED) ARE GIVEN BELOW ======)
 2800 FORMAT (///34H CARD IMAGE LISTING OF NPAR VECTOR //29X,8(I1,9X)/
     1           15H COLUMN NUMBERS,5X,8(10H1234567890)/
     2           15H NPAR VECTOR    ,5X,20A4 // )
 2750 FORMAT (//// 23H STOP  (ERRORS IN NPAR)  )
C
 2900 FORMAT (36H E L E M E N T   D E F I N I T I O N ///,
     1           14H ELEMENT TYPE ,13(2H .),16H( NPAR(1) ). . =,I5/,
     2           25H        EQ.1, TRUSS ELEMENTS/,
     3           25H        EQ.2, 2-DIM ELEMENTS/,
     4           25H        EQ.3, 3-DIM ELEMENTS/,
     5           25H        EQ.4, BEAM  ELEMENTS//,
     6           20H NUMBER OF ELEMENTS,10(2H .),16H(. NPAR(2) ). . =,I5//,
     7           28H TYPE OF NONLINEAR ANALYSIS,6(2H .),15H(. NPAR(3) ). .
     8           1H=,I5/,
     9           38H        EQ.0, LINEAR                  /,
     A           38H        EQ.1, MATERIALLY NONLINEAR ONLY  /,
     1           45H        EQ.2, UPDATED LAGRANGIAN FORMULATION       //
     2           32H ELEMENT BIRTH AND DEATH OPTIONS ,4(2H .),
     3           16H( NPAR(4) ). . =,I5/,
     4           28H        EQ.0, OPTION NOT ACTIVE/,
     5           30H        EQ.1, BIRTH OPTION ACTIVE /,
     6           30H        EQ.2, DEATH OPTION ACTIVE //)
 2920 FORMAT (18H ELEMENT TYPE CODE,11(2H .),16H( NPAR(5) ). . =,I5/,
```

```
                         TRUSS           DATE = 81229            09/18/44
      1           40H       EQ.0. GENERAL 3-D TRUSS                       /.     //.
      2           40H       EQ.1. RING ELEMENT
      4           42H MAXIMUM NUMBER OF NODES USED TO DESCRIBE  /.4X.
      5           16H ANY ONE ELEMENT.10(2H .).16H( NPAR(7) ). . =.I5//.
      6           35H INTEGRATION ORDER FOR STIFFNESS         /.4X.
      7           12H CALCULATION.12(2H .).16H( NPAR(10)). . =.I5///)
 2940 FORMAT (42H M A T E R I A L   D E F I N I T I O N    ///.
      1           16H MATERIAL MODEL..12(2H .).16H( NPAR(15)). . =.I5/.
      2           40H     EQ.1.     LINEAR ELASTIC                     /.
      3           40H     EQ.2.     NONLINEAR ELASTIC                  /.
      4           42H               (STRESS-STRAIN LAW SPECIFIED)      /.
      5           40H     EQ.3.     THERMOELASTIC                      /.
      6           44H     EQ.4.     ELASTIC-PLASTIC (ISOTROPIC)      ./.
      A           44H     EQ.5.     ELASTIC-PLASTIC (KINEMATIC)      ./.
      7           48H     EQ.6.     ELASTIC-PLASTIC-CREEP (ISOTROPIC)      /.
      8           48H     EQ.7.     ELASTIC-PLASTIC-CREEP (KINEMATIC)      /.
      8           48H     EQ.8.     USER SUPPLIED MODEL                   //.
      8           37H NUMBER OF DIFFERENT SETS OF MATERIAL            /.
      9           14H        CONSTANTS.13(2H .).16H( NPAR(16)). . =.I5//.
      B           40H NUMBER OF MATERIAL CONSTANTS PER SET. . .
      1           16H( NPAR(17)). . =.I5)
C
      END
```

```
FORTRAN IV G1  RELEASE 2.0                TRUSS                DATE = 81229

      *OPTIONS IN EFFECT*  NOTERM,ID,EBCDIC,SOURCE,NOLIST,NODECK,LOAD,NOMAP
      *OPTIONS IN EFFECT*  NAME = TRUSS    , LINECNT =        60
      *STATISTICS*     SOURCE STATEMENTS =        211,PROGRAM SIZE =      8304
      *STATISTICS*  NO DIAGNOSTICS GENERATED
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
                MAIN                    DATE = 81245             18/23/01
C          DATA SET WRITE        AT LEVEL TMP AS OF 09/02/81
C
C
C
C *CDC* *DECK WRITE
C *UNI* )FOR,IS   N.WRITE,   R.WRITE
       SUBROUTINE WRITE (DISPE,DISP,VEL,ACC,ID,NEQ,NDOE,KKK)
C
C  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C  *                                                                     *
C  *     P R O G R A M                                                   *
C  *       . TO READ INITIAL CONDITIONS INTO CORE AND                    *
C  *         PRINT THEM (IF IPRIC.EQ.1)                                  *
C  *                                                                     *
C  *       . TO PRINT DISPLACEMENTS AND (IF ISTAT.NE.0)                  *
C  *         VELOCITIES AND ACCELERATIONS                                *
C  *                                                                     *
C  *         KKK.EQ.1, READ INITIAL CONDITIONS FROM TAPE8 AND PRINT      *
C  *         KKK.EQ.2, DURING TIME INTEGRATION PRINT DISP/VEL/ACC        *
C  *                   AT NODES CONTAINED IN PRINT-OUT BLOCKS            *
C  *                                                                     *
C  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
       IMPLICIT REAL*8 (A-H,O-Z)
C
       COMMON /SOL/ NUMNP,NUMEQ,NWK,NWM,NWC,NUMEST,MIDEST,MAXEST,NSTE,MA
       COMMON /VAR/ NG,MODEX,IUPDT,KSTEP,ITEMAX,IEQREF,ITE,KPRI,
     1              IREF,IEQUIT,IPRI,KPLOTN,KPLOTE
       COMMON /PORT/ INPORT,JNPORT,NPUTSV,LUNODE,LU1,LU2,LU3,JDC,JVC,JAC
       COMMON /EL/ IND,ICOUNT,NPAR(20),NUMEG,NEGL,NEGNL,IMASS,IDAMP,ISTAT
     1              ,NDOFDM,KLIN,IEIG,IMASSN,IDAMPN
       COMMON /PRCON/ IDAT,R,IPRIC,NPB,IDC,IVC,IAC,IPC,IPNODE(3,15)
       COMMON /CONST/ DT,DTA,A0,A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11
     1              ,A12,A13,A14,A15,A16,A17,A18,A19,A20,IOPE
       COMMON /FORCEB/ F1P(100),F2P(100),FR(100)
       COMMON /ELSTP/ TIME,IDTHF
       COMMON /PLOTE/ MB,NR(100),NS(100)
       COMMON /MDFRDM/ IDOF(6)
       DIMENSION DISPE(NEQ),DISP(NEQ),VEL(NEQ),ACC(NEQ),ID(NDOF,1)
       DIMENSION D(6)
C
       IF (KSTEP.NE.0) GO TO 59
       DO 10 IP=1,MB
       F1P(IP)=0.
       F2P(IP)=0.
    10 FR(IP) =0.
    59 DO 25 IP=1,MB
    25 WRITE (20,30) KSTEP,TIME,F1P(IP),F2P(IP),FR(IP)
    30 FORMAT (I10,4D13.6)
C      READ ID ARRAY INTO CORE
C
       REWIND 8
       READ(8) ((ID(I,J),I=1,NDOF),J=1,NUMNP)
       IF (KKK.GT.1) GO TO 50
C
C      READ INITIAL CONDITIONS INTO CORE
C
       READ (8) DISP
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
      IF (ISTAT.EQ.0) GO TO 40
      IF (IOPE.EQ.3) GO TO 20
      READ (8) VEL
      READ (8) ACC
      GO TO 40
C
   20 READ (8) DISPE
      IF (MODEX.EQ.2) GO TO 40
      ISV=(IVC + JVC + 1)/2
      ISA=(IAC + JAC + 1)/2
      IF (ISV.EQ.0) READ (8)
      IF (ISV.NE.0) READ (8) VEL
      IF (ISA.NE.0) READ (8) ACC
C
   40 IF (IPRIC.EQ.0) RETURN
      WRITE (6,2100)
C
C     PRINT DISPLACEMENTS
C
   50 IC=4
      IF (KKK.EQ.1) GO TO 60
      IF (IDC.EQ.0) GO TO 180
   60 WRITE(6,2000)
      IC=IC + 5
      DO 150 IB=1,NRB
      NODE1=IPNODE(1,IB)
      IF (NODE1.EQ.0) GO TO 150
      NODE2=IPNODE(2,IB)
      NODINC=IPNODE(3,IB)
      IF (KKK.EQ.1) NODE1=1
      IF (KKK.EQ.1) NODE2=NUMNP
      IF (KKK.EQ.1) NODINC=1
C
      DO 100 II=NODE1,NODE2,NODINC
      IC=IC + 1
      IF (IC.LT.56) GO TO 105
      WRITE(6,2045)
      IC=4
  105 DO 110 I=1,6
  110 D(I)=0.
      IL=0
      DO 120 I=1,NDOF
      KK=LD(I,II)
  115 IL=IL + 1
      IF (IL.LE.6) GO TO 117
      WRITE (6,3000)
      STOP
  117 IF (IDOF(IL).EQ.1) GO TO 115
      IF (KK.NE.0) D(IL)=DISP(KK)
      IF (IOPE.NE.3.OR.IND.NE.0) GO TO 120
      IF (KK.NE.0) D(IL)=DISPE(KK)
  120 CONTINUE
      DO 1000 IX=1,MB
      IF (II.EQ.NR(IX).OR.II.EQ.NS(IX)) WRITE(20,5000) II,D(2),D(3)
 1000 CONTINUE
 5000 FORMAT (I10,2D13.5)
  100 WRITE(6,2010) II,D
C
```

ORIGINAL PAGE IS
OF POOR QUALITY

WRITE DATE = 81245 18/23/01

```
      IF (KKK.EQ.1) GO TO 180
      IF (IC.GE.55) GO TO 150
      IC=IC+1
      WRITE(6,2050)
  150 CONTINUE
  180 IF (ISTAT.EQ.0) RETURN
C
C     PRINT VELOCITIES
C
      IF (KKK.EQ.1 .AND. IOPE.NE.3) GO TO 201
      IF (IVC.EQ.0) GO TO 280
  201 IC=IC + 5 + IDC
      IF (IDC.NE.0) WRITE(6,2050)
      IF (IC.GE.54) GO TO 205
      WRITE(6,2020)
      GO TO 206
  205 WRITE(6,2022)
      IC=4
  206 DO 250 IB=1,NPB
      NODE1=IPNODE(1,IB)
      IF (NODE1.EQ.0) GO TO 250
      NODE2=IPNODE(2,IB)
      NODINC=IPNODE(3,IB)
      IF (KKK.EQ.1) NODE1=1
      IF (KKK.EQ.1) NODE2=NUMNP
      IF (KKK.EQ.1) NODINC=1
C
      DO 200 II=NODE1,NODE2,NODINC
      IC=IC + 1
      IF (IC.LT.56) GO TO 207
      WRITE(6,2022)
      IC=4
  207 DO 210 I=1,6
  210 D(I)=0.
      IL=0
      DO 220 I=1,NDOF
      KK=ID(I,II)
  215 IL=IL + 1
      IF (IL.LE.6) GO TO 217
      WRITE (6,3000)
      STOP
  217 IF (IDOF(IL).EQ.1) GO TO 215
  220 IF (KK.NE.0) D(IL)=VEL(KK)
      DO 1001 IX=1,MB
      IF (II.EQ.NH(IX).OR.II.EQ.NS(IX)) WRITE(20,5000) II,D(2),D(3)
 1001 CONTINUE
  200 WRITE(6,2010) II,D
C
      IF (KKK.EQ.1) GO TO 280
      IF (IC.GE.55) GO TO 250
      IC=IC+1
      WRITE(6,2050)
  250 CONTINUE
C
C     PRINT ACCELERATIONS
C
  280 IF (KKK.EQ.1 .AND. IOPE.NE.3) GO TO 290
      IF (IAC.EQ.0) RETURN
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
      IF (IDC.EQ.0 .AND. IVC.EQ.0) GO TO 305
  290 IC=IC + 6
      IF (IC.GE.54) GO TO 303
      WRITE(6,2050)
      WRITE(6,2030)
      GO TO 308
  303 WRITE(6,2032)
      IC=4
      GO TO 308
  305 IC=IC + 5
      WRITE(6,2030)
  308 DO 350 IB=1,NPB
      NODE1=IPNODE(1,IB)
      IF (NODE1.EQ.0) GO TO 350
      NODE2=IPNODE(2,IB)
      NODINC=IPNODE(3,IB)
      IF (KKK.EQ.1) NODE1=1
      IF (KKK.EQ.1) NODE2=NUMNP
      IF (KKK.EQ.1) NODINC=1
C
      DO 300 II=NODE1,NODE2,NODINC
      IC=IC + 1
      IF (IC.LT.56) GO TO 307
      WRITE(6,2032)
      IC=4
  307 DO 310 I=1,6
  310 D(I)=0.
      IL=0
      DO 320 I=1,NDOF
      KK=ID(I,II)
  315 IL=IL + 1
      IF (IL.LE.6) GO TO 317
      WRITE (6,3000)
      STOP
  317 IF (IDOF(IL).EQ.1) GO TO 315
  320 IF (KK.NE.0) D(IL)=ACC(KK)
      DO 1002 IX=1,MB
      IF (II.EQ.NR(IX).OR.II.EQ.NS(IX)) WRITE(20,5000) II,D(2),D(3)
 1002 CONTINUE
  300 WRITE(6,2010) II,D
C
      IF (KKK.EQ.1) RETURN
      IF (IC.GE.55) GO TO 350
      IC=IC+1
      WRITE(6,2050)
  350 CONTINUE
C
      RETURN
C
 2000 FORMAT (/27H  D I S P L A C E M E N T S  // 7H  NODE 12X
     114HX-DISPLACEMENT 4X 14HY-DISPLACEMENT 4X 14HZ-DISPLACEMENT
     28X 10HX-ROTATION 8X 10HY-ROTATION 8X 10HZ-ROTATION /)
 2010 FORMAT (2X,I5,8X,6E18.6)
 2020 FORMAT(/22H  V E L O C I T I E S  // 7H  NODE 16X 10HX-VELOCITY
     18X 10HY-VELOCITY 8X 10HZ-VELOCITY
     28X 10HX-ROTATION 8X 10HY-ROTATION 8X 10HZ-ROTATION /)
 2022 FORMAT (1H1,21H V E L O C I T I E S  //7H  NODE 16X 10HX-VELOCITY
     18X 10HY-VELOCITY 8X 10HZ-VELOCITY 8X 10HX-ROTATION 8X
```

WRITE DATE = 81245 18/23/01

```
     310HY=ROTATION 8X 10HZ=ROTATION  /)
2030 FORMAT (/27H  A C C E L E R A T I O N S  // 7H  NODE 12X
    114HX=ACCELERATION 4X 14HY=ACCELERATION 4X 14HZ=ACCELERATION
     28X 10HX=ROTATION 8X 10HY=ROTATION 8X 10HZ=ROTATION  /)
2032 FORMAT (1H1,26H A C C E L E R A T I O N S  // 7H  NODE 12X
    114HX=ACCELERATION 4X 14HY=ACCELERATION 4X 14HZ=ACCELERATION
     28X 10HX=ROTATION 8X 10HY=ROTATION 8X 10HZ=ROTATION  /)
2042 FORMAT (1H1, 28H D I S P L A C E M E N T S  // 7H  NODE 12X
    114HX=DISPLACEMENT 4X 14HY=DISPLACEMENT 4X 14HZ=DISPLACEMENT
     28X 10HX=ROTATION 8X 10HY=ROTATION 8X 10HZ=ROTATION  /)
2050 FORMAT (1H )
2100 FORMAT (1H1,38H I N I T I A L   C O N D I T I O N S     ///)
3000 FORMAT (///48H **STOP, ERROR IN DEGREE OF FREEDOM CALCULATIONS,/,
    1            28H CHECK MASTER CONTROL CARD 1   ,/1X)
C
      END
```

WRITE DATE = 81245

```
*OPTIONS IN EFFECT*  NOTERM,ID,EBCDIC,SOURCE,NOLIST,NODECK,LOAD,NOMAP,NOTEST
*OPTIONS IN EFFECT*  NAME = WRITE    , LINECNT =          60
*STATISTICS*     SOURCE STATEMENTS =        183,PROGRAM SIZE =        5976
*STATISTICS*  NO DIAGNOSTICS GENERATED

*STATISTICS*  NO DIAGNOSTICS THIS STEP
```

APPENDIX 4: TWO-DIMENSIONAL PLOT CODE

During the solution phase, the main program "ADINA" solves for the responses of the bearing structural problem incrementally in time. These responses, namely the displacements, velocities, accelerations, and bearing forces, are stored on an out-of-core tape for future plotting purposes. The subroutine "WRITE" of "ADINA" has been modified to store on a tape the responses of the rotors and stators of squeeze-film dampers of either a single bearing or multi-bearing problems.

The code listed below reads and plots the rotor and stator displacement, velocity, and/or acceleration trajectories of any squeeze-film damper. For any desired plot, separate graphs for either the stator or the rotor responses can be obtained. For the displacement, the difference between the rotor and stator displacements are plotted to show the rotor orbit relative to the stator. For the bearing forces, both the actual force and the filtered force plots can be obtained. A linear filtering, by averaging the forces, has been used here. However, other methods of filtering can be employed by the user.

To help increase the diversity and capabilities of the code, different control flags or parameters have been introduced to give the user the opportunity to turn on or shut off any graph. For example, KPD is a flag that indicates displacement plotting mode. A value of "1" for KPD will request the code to plot displacements of both the rotor and stator.

The following control parameters must be input by the user:

KPD: Control parameter indicating plotting mode;

EQ.1: plot rotor and stator displacement trajectories

EQ.0: plots are not required

KPV: Control parameter indicating velocity plotting mode;

EQ.1: plot rotor and stator velocity trajectories

EQ.0: plots are not required

KPA: Control parameter indicating acceleration plotting mode;

EQ.1: plot rotor and stator acceleration trajectories

EQ.0: plots are not required.

ORIGINAL PAGE IS
OF POOR QUALITY


```
 1   //CASEN1     JOB   XXXXX,'3523 ZEID                    ',MSGLEVEL=(2,0)
     ***JOBPARM        SKIP=YES,FORMS=1PWB
 2   // EXEC FORT
 9   //SYSIN       DD *                      GENERATED STATEMENT
10   // EXEC GOFORT,PLOT=,GOSIZE=900K
31   //SYSIN       DD *                      GENERATED STATEMENT
32   //FT23F001    DD   DSN=USER.PROBN1,DISP=(OLD,KEEP)
33   //SYSIN       DD *                      GENERATED STATEMENT
     //
```

```
IEF142I CASEN1 FORT - STEP WAS EXECUTED - COND CODE 0000
IEF373I STEP /FORT     / START 81232.2330
IEF374I STEP /FORT     / STOP  81232.2331 CPU    0MIN 05.55SEC SRB
IEF142I CASEN1 GO - STEP WAS EXECUTED - COND CODE 0000
IEF373I STEP /GO       / START 81232.2331
IEF374I STEP /GO       / STOP  81232.2334 CPU    0MIN 53.21SEC SRB
IEF375I  JOB /CASEN1   / START 81232.2330
IEF376I  JOB /CASEN1   / STOP  81232.2334 CPU    0MIN 58.76SEC SRB
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
                    MAIN                    DATE = 81232                23/30/52
      DIMENSION DXR(2000),DYR(2000),DXS(2000),DYS(2000),VXR(2000),
     1          VYR(2000),VXS(2000),VYS(2000),AXR(2000),AYR(2000),
     2          AXS(2000),AYS(2000),XARRAY(4500),YARRAY(4500),NPTS(4),
     3          INC(4),LINTYP(4),INTEQ(4),NR(100),NS(100)
      DIMENSION F1(2000),F2(2000),FR(2000),TIM(2000)
      DOUBLE PRECISION D(3),F1P(100),F2P(100),FRP(100),TIME
C
C
C     **************************************************************
C     *                                                            *
C     * THIS CODE PLOTS THE ROTOR AND STATOR DISPLACEMENT,VELOCITY, *
C     *                                                            *
C     * AND/OR ACCELERATION TRAJECTORIES OF SQUEEZ-FILM DAMPER. IT ALSO *
C     *                                                            *
C     * PLOTS THE TRAJECTORIES OF  THE BEARING FORCES. THESE RESULTS *
C     *                                                            *
C     * ARE GENERATED AND STORED ON A TAPE BY THE MAIN CODE **ADINA** *
C     *                                                            *
C     * DURING THE SOLUTION PHASE.                                 *
C     *                                                            *
C     * ************                                               *
C     *                                                            *
C     * ************                                               *
C     *                                                            *
C     * NOMENCLATURE                                               *
C     *                                                            *
C     * ************                                               *
C     *                                                            *
C     * ************                                               *
C     *                                                            *
C     * KPD : CONTROL PARAMETER INDICATING DISPLACEMENT PLOTTING MODE: *
C     *                                                            *
C     *                EQ. 1: PLOT ROTOR AND STATOR DISP. TRAJECTORIES *
C     *                                                            *
C     *                EQ. 0: PLOTS ARE NOT REQUIRED               *
C     *                                                            *
C     * KPV : CONTROL PARAMETER INDICATING VELOCITY PLOTTING MODE: *
C     *                                                            *
C     *                EQ. 1: PLOT ROTOR AND STATOR VEL. TRAJECTORIES *
C     *                                                            *
C     *                EQ. 0: PLOTS ARE NOT REQUIRED               *
C     *                                                            *
C     * KPA : CONTROL PARAMETER INDICATING ACCELERATION PLOTTING MODE: *
C     *                                                            *
C     *                EQ. 1: PLOT ROTOR AND STATOR ACC. TRAJECTORIES *
C     *                                                            *
C     *                EQ. 0: PLOTS ARE NOT REQUIRED               *
C     *                                                            *
C     * KPF : CONTROL PARAMETER INDICATING FORCES PLOTTING MODE:   *
C     *                                                            *
C     *                EQ. 1: PLOT TRAJECTORIES OF BEARING FORCES  *
C     *                                                            *
C     *                EQ. 0: PLOTS ARE NOT REQUIRED               *
C     *                                                            *
C     * DXR,VXR,AXR : ARRAYS WHERE THE ROTOR HORIZONTAL DISPLACEMENT, *
C     *                                                            *
```

```
C     *                    VELOCITY, AND ACCELERATION COMPONENTS ARE           *
C     *                                                                        *
C     *                    STORED RESPECTIVELY.                                *
C     *                                                                        *
C     * DYR,VYR,AYR : ARRAYS WHERE THE ROTOR VERTICAL   DISPLACEMENT,          *
C     *                                                                        *
C     *                    VELOCITY, AND ACCELERATION COMPONENTS ARE           *
C     *                                                                        *
C     *                    STORED RESPECTIVELY.                                *
C     *                                                                        *
C     * DXS,VXS,AXS : ARRAYS WHERE THE STATOR HORIZONTAL DISPLACEMENT,         *
C     *                                                                        *
C     *                    VELOCITY, AND ACCELERATION COMPONENTS ARE           *
C     *                                                                        *
C     *                    STORED RESPECTIVELY.                                *
C     *                                                                        *
C     * DYS,VYS,AYS : ARRAYS WHERE THE STATOR VERTICAL   DISPLACEMENT,         *
C     *                                                                        *
C     *                    VELOCITY, AND ACCELERATION COMPONENTS ARE           *
C     *                                                                        *
C     *                    STORED RESPECTIVELY.                                *
C     *                                                                        *
C     * F1,F2 : ARRAYS WHERE HORIZONTAL AND VERTICAL BEARING FORCE             *
C     *         COMPONENTS ARE STORED                                          *
C     *                                                                        *
C     * FR : ARRAY WHERE THE RESULTANT BEARING FORCES ARE STORED.              *
C     *                                                                        *
C     *                                                                        *
C     **************************************************************************
C
C
C
      K=1
      KPD=1
      KPV=1
      KPA=0
      KPF=1
      BC=0.01
      M=2*K
      IR=23
      N=1999
C
C     RAED ROTOR AND STATOR NODE NUMBERS
C
      READ (5,1) ((NR(I),NS(I)),I=1,K)
      DO 504 KK=1,K
      WRITE (6,505) NR(KK),NS(KK)
C
C     READ BEARING FORCES INTO CORE
C
      DO 500 I=1,N
      DO 9 IP=1,K
      READ (IR,6) KSTEP,TIME,F1P(IP),F2P(IP),FRP(IP)
      IF (KPF.NE.1) GO TO 2
      WRITE (6,7) KSTEP,TIME,F1P(IP),F2P(IP),FRP(IP)
C
C     MOVE BEARING FORCES INTO ARRAYS F1,F2, AND FR FOR PLOTTING PURPOSES
C
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
      TIM(I)=TIME
      F1(I+(IP-1)*N)=F1P(IP)
      F2(I+(IP-1)*N)=F2P(IP)
    9 FR(I+(IP-1)*N)=FRP(IP)
C
C     READ ROTOR AND STATOR DISPLACEMENTS INTO CORE
C
    2 DO 501 J=1,M
      READ (IR,5) II,D(2),D(3)
C     WRITE (6,5) II,D(2),D(3)
      IF (KPD.NE.1) GO TO 501
      DO 401 L=1,K
      IF (II.EQ.NR(L)) DXR(I+(L-1)*N)=D(2)
      IF (II.EQ.NR(L)) DYR(I+(L-1)*N)=D(3)
      IF (II.EQ.NS(L)) DXS(I+(L-1)*N)=D(2)
      IF (II.EQ.NS(L)) DYS(I+(L-1)*N)=D(3)
  401 CONTINUE
  501 CONTINUE
C
C     READ ROTOR AND STATOR VELOCITIES INTO CORE
C
      DO 502 J=1,M
      READ (IR,5) II,D(2),D(3)
C     WRITE (6,5) II,D(2),D(3)
      IF (KPV.NE.1) GO TO 502
      DO 402 L=1,K
      IF (II.EQ.NR(L)) VXR(I+(L-1)*N)=D(2)
      IF (II.EQ.NR(L)) VYR(I+(L-1)*N)=D(3)
      IF (II.EQ.NS(L)) VXS(I+(L-1)*N)=D(2)
      IF (II.EQ.NS(L)) VYS(I+(L-1)*N)=D(3)
  402 CONTINUE
  502 CONTINUE
C
C     READ ROTOR AND STATOR ACCELERATIONS INTO CORE.
C
      DO 503 J=1,M
      READ (IR,5) II,D(2),D(3)
C     WRITE (6,5) II,D(2),D(3)
      IF (KPA.NE.1) GO TO 503
      DO 403 L=1,K
      IF (II.EQ.NR(L)) AXR(I+(L-1)*N)=D(2)
      IF (II.EQ.NR(L)) AYR(I+(L-1)*N)=D(3)
      IF (II.EQ.NS(L)) AXS(I+(L-1)*N)=D(2)
      IF (II.EQ.NS(L)) AYS(I+(L-1)*N)=D(3)
  403 CONTINUE
  503 CONTINUE
  500 CONTINUE
  504 CONTINUE
      DO 510 L=1,K
      IF (KPD.NE.1) GO TO 507
      WRITE (6,10) (DXR(I+(L-1)*N),I=1,N)
      WRITE (6,11) (DYR(I+(L-1)*N),I=1,N)
      WRITE (6,12) (DXS(I+(L-1)*N),I=1,N)
      WRITE (6,13) (DYS(I+(L-1)*N),I=1,N)
  507 IF (KPV.NE.1) GO TO 506
      WRITE (6,14) (VXR(I+(L-1)*N),I=1,N)
      WRITE (6,15) (VYR(I+(L-1)*N),I=1,N)
      WRITE (6,16) (VXS(I+(L-1)*N),I=1,N)
```

```
      WRITE (6,17) (VYS(I+(L-1)*N),I=1,N)
  506 IF (KPA.NE.1) GO TO 510
      WRITE (6,13) (AXR(I+(L-1)*N),I=1,N)
      WRITE (6,19) (AYR(I+(L-1)*N),I=1,N)
      WRITE (6,20) (AXS(I+(L-1)*N),I=1,N)
      WRITE (6,21) (AYS(I+(L-1)*N),I=1,N)
  510 CONTINUE
      CALL PLOTS
      CALL PLOT (1.0,1.5,-3)
      DO 610 L=1,K
      NPTS(1)=N
      NPTS(2)=N
      INC(1)=1
      INC(2)=1
      LINTYP(1)=35
      LINTYP(2)=35
      INTEQ(1)=4
      INTEQ(2)=0
      IF (KPD.NE.1) GO TO 22
C
C     PLOT ROTOR DISPLACEMENT TRAJECTORIES
C
      DO 550 I=1,N
      XARRAY(I)=DXR(I+(L-1)*N)
  550 YARRAY(I)=DYR(I+(L-1)*N)
      IF (L.NE.1) CALL PLOT(12.0,0.0,-3)
      CALL GRAPH (XARRAY,'Y-DISP',6,7.5,YARRAY,' Z-DISP',6,7.5,
     1'ROTOR DISPLACEMENT TRAJECTORIES',31,0.5,7.75,0.14,1,
     2      NPTS,INC,LINTYP,INTEQ)
C
C     PLOT STATOR DISPLACEMENT TRAJECTORIES
C
      DO 551 I=1,N
      XARRAY(I)=DXS(I+(L-1)*N)
  551 YARRAY(I)=DYS(I+(L-1)*N)
      CALL PLOT (12.0,0.0,-3)
      CALL GRAPH (XARRAY,'Y-DISP',6,7.5,YARRAY,' Z-DISP',6,7.5,
     1'STATOR DISPLACEMENT TRAJECTORIES',32,0.5,7.75,0.14,1,
     2      NPTS,INC,LINTYP,INTEQ)
C
C     PLOT ROTOR ORBIT RELATIVE TO STATOR
C
      DO 561 I=1,N
      XARRAY(I)=DXR(I+(L-1)*N)-DXS(I+(L-1)*N)
  561 YARRAY(I)=DYR(I+(L-1)*N)-DYS(I+(L-1)*N)
      DTH=3.141593/90.0
      DO 552 I=1,180
      I2=I+N+2
      XARRAY(I2)=BC*COS(DTH*(I-1))
  552 YARRAY(I2)=BC*SIN(DTH*(I-1))
      XARRAY(181+N+2)=XARRAY(N+3)
      YARRAY(181+N+2)=YARRAY(N+3)
      NPTS(2)=181
      LINTYP(1)=0
      LINTYP(2)=0
      CALL PLOT (12.0,0.0,-3)
      CALL GRAPH (XARRAY,'Y-DISP',6,7.5,YARRAY,' Z-DISP',6,7.5,
     1'ROTOR ORBIT',11,0.5,7.75,0.14,2,
```

MAIN DATE = 81232

```
      2       NPTS,INC,LINTYP,INTEQ)
   22 IF (KPV.NE.1) GO TO 23
C
C     PLOT ROTOR VELOCITY TRAJECTORIES
C
      DO 580 I=1,N
      XARRAY(I)=VXR(I+(L-1)*N)
  580 YARRAY(I)=VYR(I+(L-1)*N)
      NPTS(2)=N
      LINTYP(1)=35
      LINTYP(2)=35
      CALL PLOT (12.0,0.0,-3)
      CALL GRAPH (XARRAY,'Y-VEL ',6,7.5,YARRAY,' Z-VEL ',6,7.5,
     1'ROTOR VELOCITY TRAJECTORIES',27,0.5,7.75,0.14,2,
     2       NPTS,INC,LINTYP,INTEQ)
C
C     PLOT STATOR VELOCITY TRAJECTORIES
C
      DO 579 I=1,N
      XARRAY(I)=VXS(I+(L-1)*N)
  579 YARRAY(I)=VYS(I+(L-1)*N)
      CALL PLOT (12.0,0.0,-3)
      CALL GRAPH (XARRAY,'Y-VEL ',6,7.5,YARRAY,' Z-VEL ',6,7.5,
     1'STATOR VELOCITY TRAJECTORIES',28,0.5,7.75,0.14,2,
     2       NPTS,INC,LINTYP,INTEQ)
   23 IF (KPA.NE.1) GO TO 24
C
C     PLOT ROTOR ACCELERATION TRAJECTORIES
C
      DO 600 I=1,N
      XARRAY(I)=AXR(I+(L-1)*N)
  600 YARRAY(I)=AYR(I+(L-1)*N)
      CALL PLOT (12.0,0.0,-3)
      CALL GRAPH (XARRAY,'Y-ACC ',6,7.5,YARRAY,' Z-ACC ',6,7.5,
     1'ROTOR ACCELERATION TRAJECTORIES',31,0.5,7.75,0.14,2,
     2       NPTS,INC,LINTYP,INTEQ)
C
C     PLOT STATOR ACCELERATION TRAJECTORIES
C
      DO 590 I=1,N
      XARRAY(I)=AXS(I+(L-1)*N)
  590 YARRAY(I)=AYS(I+(L-1)*N)
      CALL PLOT (12.0,0.0,-3)
      CALL GRAPH (XARRAY,'Y-ACC ',6,7.5,YARRAY,' Z-ACC ',6,7.5,
     1'STATOR ACCELERATION TRAJECTORIES',32,0.5,7.75,0.14,2,
     2       NPTS,INC,LINTYP,INTEQ)
   24 IF (KPF.NE.1) GO TO 610
C
C     PLOT HORIZONTAL COMPONENT OF BEARING FORCE
C
      DO 605 I=1,N
      XARRAY(I)=TIM(I)
  605 YARRAY(I)=F1(I+(L-1)*N)
      CALL PLOT(12.0,0.0,-3)
      CALL GRAPH (XARRAY,'TIME',4,7.5,YARRAY,'ROTOR FORCE(HORIZONTAL)',
     1             23,7.5,'ROTOR FORCE TRAJECTORIES',24,0.5,7.75,0.14,
     2             1,NPTS,INC,LINTYP,INTEQ)
```

ORIGINAL PAGE IS
OF POOR QUALITY

MAIN DATE = 81232

```
C
C     PLOT VERTICAL COMPONENT OF BEARING FORCE
C
      DO 606 I=1,N
  606 YARRAY(I)=F2(I+(L-1)*N)
      CALL PLOT (12.0,0.0,-3)
      CALL GRAPH (XARRAY,'TIME',4,7.5,YARRAY,'ROTOR FORCE(VERTICAL)',
     1            21,7.5,'ROTOR FORCE TRAJECTORIES',24,0.5,7.75,0.14,
     2            1,NPTS,INC,LINTYP,INTEQ)
C
C     PLOT FILTERED HORIZONTAL COMPONENT OF BEARUNG FORCE:
C     (LINEAR FILTERING: F(I)=(F(I)+F(I+1))/2  )
C
      N1=N-1
      DO 581 I=1,N1
      XARRAY(I)=(TIM(I)+TIM(I+1))/2.0
  581 YARRAY(I)=(F1(I+(L-1)*N)+F1(I+1+(L-1)*N))/2.0
      NPTS(1)=N1
      CALL PLOT (12.0,0.0,-3)
      CALL GRAPH (XARRAY,'TIME',4,7.5,YARRAY,'ROTOR FORCE(HORIZONTAL)',
     1 23,7.5,'FILTERED FORCE TRAJECTORIES',27,0.5,7.75,0.14,
     2            1,NPTS,INC,LINTYP,INTEQ)
C
C     PLOT FILTERED VERTICAL COMPONENT OF BEARING FORCE
C     (  LINEAR FILTERING  )
C
      DO 582 I=1,N1
  582 YARRAY(I)=(F2(I+(L-1)*N)+F2(I+1+(L-1)*N))/2.0
      CALL PLOT (12.0,0.0,-3)
      CALL GRAPH (XARRAY,'TIME',4,7.5,YARRAY,'ROTOR FORCE(VERTICAL)',
     1 23,7.5,'FILTERED FORCE TRAJECTORIES',27,0.5,7.75,0.14,
     2            1,NPTS,INC,LINTYP,INTEQ)
  610 CONTINUE
      CALL PLOT (12.0,0.0,999)
    1 FORMAT (16I5)
    5 FORMAT (I10,2D13.6)
    6 FORMAT (I10,4D13.6)
    7 FORMAT (50X,I10,4(2X,D13.6))
   10 FORMAT (///5X,'HORIZONTAL ROTOR DISPLACEMENT',//,8(2X,E13.5))
   11 FORMAT (///5X,'VERTICAL   ROTOR DISPLACEMENT',//,8(2X,E13.5))
   12 FORMAT (///5X,'HORIZONTAL STATR DISPLACEMENT',//,8(2X,E13.5))
   13 FORMAT (///5X,'VERTICAL   STATR DISPLACEMENT',//,8(2X,E13.5))
   14 FORMAT (///5X,'HORIZONTAL  ROTOR VELOCITY:',//,8(2X,D13.5))
   15 FORMAT (///5X,'VERTICAL  ROTOR VELOCITY:',//,8(2X,D13.5))
   16 FORMAT (///5X,'HORIZONTAL STATOR VELOCITY:',//,8(2X,D13.5))
   17 FORMAT (///5X,'VERTICAL STATOR VELOCITY:',//,8(2X,D13.5))
   18 FORMAT (///5X,'HORIZONTAL  ROTOR ACCELERATION:',//,8(2X,D13.5))
   19 FORMAT (///5X,'VERTICAL  ROTOR ACCELERATION:',//,8(2X,D13.5))
   20 FORMAT (///5X,'HORIZONTAL STATOR ACCELERATION:',//,8(2X,D13.5))
   21 FORMAT (///5X,'VERTICAL STATOR ACCELERATION:',//,8(2X,D13.5))
  505 FORMAT (1H0///,5X,'P R I N T O U T   F O R   R O T O R  ',I5,
     1          ' A N D   S T A T O R   ',I5,///)
      STOP
      END
```

MAIN DATE = 81232

```
*OPTIONS IN EFFECT*  NOTERM,ID,EBCDIC,SOURCE,NOLIST,NODECK,LOAD,NOMAP
*OPTIONS IN EFFECT*  NAME = MAIN    , LINECNT =       60
*STATISTICS*     SOURCE STATEMENTS =      177,PROGRAM SIZE =    175892
*STATISTICS*  NO DIAGNOSTICS GENERATED
```

C 2
3

APPENDIX 5: THREE-DIMENSIONAL PLOTTING CODE

This Appendix overviews the post-processor programs written to plot three dimensional pictures of a deformed shaft given the deflected positions of distinct points along the shaft at a sequence of time intervals. The purpose of such plots is to reduce computed rotor vibration events (transient as well as steady-state) to an easily visualized series of "still" pictures. A sequence of such pictures is then suitable to construct a slow-motion animation of rotor vibration phenomena. This Appendix provides a detailed description of the necessary inputs to the post-processor programs as well as describing the post-processor programs themselves.

A5.1 Post-Processor 1, JCL Requirements

There are three JCL requirements for the first of the two post-processor programs. The first of these three requirements is the allocation of FORTRAN logical unit 20 in the modified ADINA program to provide input to the post-processor program. Execution of this post-processor would be simplified if the data that is to be read by this unit exists on the disk data storage of the computer. If this data does exist on a disk dataset, then the JCL necessary for unit 20 is that which will allow this unit to access the appropriate disk dataset.

The second JCL requirement of the first post-processor program deals with the datasets that are created during the execution of this program that facilitate the plotting of

the orbit trajectory. During the execution of this post-processor program, the time-displacement data for all the nodes along the shaft are read and then this data is written in a manner that a dataset is created for each node along the shaft. Each dataset that is created contains only the time-displacement data for one node. The units associated with this manner of data manipulation begin with FORTRAN logical unit 11 and continue through unit 11 plus the number of nodes along the shaft. Each of these units will require the JCL to assign them to different disk datasets that are created during the program. Per the above discussion, if there are 4 nodes present along the shaft, then the JCL supplied will have to define FORTRAN logical units 11, 12, 13, and 14.

The third JCL requirement of the first post-processor program concerns the dataset which contains the output from this segment of the plotting package. This output is the coefficients of the interpolatory polynomial. This data is output to a disk dataset so that it can be retrieved by the second post-processor program. The FORTRAN logical unit associated with this output is unit 21. Therefore, the JCL to create and save a dataset that will be allocated to unit 21 must be provided.

A5.2 Post-Processor 2, JCL Requirements

The JCL requirements for this post-processor have been alluded to in the previous section. These are twofold and both deal with datasets that are created by the first post-processor program.

First, beginning with FORTRAN logical unit 11 and assigning one unit for each node, the datasets that contain the orbit trajectory informa-

tion must be allocated to FORTRAN logical unit. This allocation procedure must be similar to the allocations that were made in executing the first post-processor. This means that the dataset allocated to unit 11 during execution of the first post-processor must be allocated to unit 11 during the execution of the second post-processor.

The second JCL requirement of this post-processor program is the allocation of the dataset (created during the execution of the first post-processor program) which contains the output from the first post-processor. This dataset must be allocated to FORTRAN logical unit 21 during the execution of the second post-processor program.

A5.3 Inputs from Modified ADINA Code

The output from the modified ADINA program serves as input to the first post-processor program contains four different types of records. These records and the order in which they should appear with their formats are as follows:

(1) Number of nodes which appear along the shaft in a (2X, I10) format.

(2) X, Y, and Z static, undeformed position of these nodes in a (3(2X, E15.8) format. Note, these nodes should be ordered so that the node closest to the origin appears first and the node farthest away from the origin along the shaft axis appears last.

(3) The period of shaft revolution in seconds per cycle in a (2X, E15.7) format.

(4) The node id, time, X, Y, and Z displacement for each node at each time step of integration in a (I10, 4E13.6) format.

The records defined by (1), (2), and (3) will appear once. However, the records defined by item (4) will appear for each node for every time step during the solution of the analysis.

A5.4 User Inputs

A5.4A User Inputs to Post-Processor I

A5.4A.1 Card No. 1

This card defines the times at which the deformed shaft is to be plotted.

1 10	11 20	21 30	
ANGINT	ANGFIN	ANGINC	(3F10.0)

ANGINT - Value of the initial revolution at which the deformed shaft is to be plotted

ANGFIN - Value of the final revolution at which the deformed shaft is to be plotted

ANGINC - Value of the revolution increment at which the deformed shaft is to be plotted. The time of the i^{th} plot will be:

TIME = (ANGINT + (i-1)*ANGINC)*PERIOD

Where: PERIOD = period of shaft revolution.

A5.4A.2 Card No. 2

This card defines the direction of the shaft axis.

1	
I	(1A1)

Where: I is an alphanumeric X, Y, or Z depending upon the shaft orientation.

Note: For X1, X2, X3 coordinate systems;

X = X1

Y = X2

Z = X3

A5.4B User Inputs to Post-Processor 2

A5.4B.1 Card No. 1

1 5	6 15	
NSECT	THETA	

(I5,F10.0)

Where: NSECT - The number of sections into which the shaft will be divided for the plotting of its deformed shape. A high value for this number will cause a large CPU time for the plotting procedure. However, a low value for this number will cause the plot to be choppy in appearance.

THETA - The angle at which the shaft axis will be inclined from the horizontal.

A5.4B.2 Card No. 2

This card reads the plot control parameters. The control parameters consist of 6 integers read in a (I5, 5I1) format. The first integer controls the number of deformed shafts that are drawn on each plot. The second integer, read from the sixth card column, controls the usage of the default plot window size. If this parameter is input

as a zero or blank, then the plot drawn will fit in an eleven inch square. If this parameter is input as a one, then the plot size will be read from input card number 3. The third control integer, read from the seventh card column, controls the maximum size of the orbit trajectory. If this parameter is input as a zero or blank, then the maximum orbit trajectory will be scaled to one and one-half inches on the plot. If this parameter is input as a one, then the plot size of the maximum orbit trajectory will be read from input card number 4. The fourth control integer, read from the eighth card column, controls the appearance of the node names on the plot. If this parameter is input as a zero or a blank, then the node names will appear on the plot. If this parameter is input with a value of one, then the node names do not appear. The fifth control parameter, read from the ninth card column, controls the appearance of the line connecting a node's static, undeformed, position with its deformed position. If this parameter is input as zero or blank, then this line appears. If this parameter is input as a one, then this line does not appear on the plot. The final control parameter, read from card column number ten, controls the time range over which the orbit trajectory is plotted. If this parameter is input as zero or blank, then the entire orbit trajectory is plotted. If this parameter is input as a one, then the time range over which the orbit is plotted is defined by the data appearing on card number 5.

Summary:

Parameter	Controls
1	Number of shafts per picture
2	Plot size
3	Orbit trajectory size
4	Node name appearance
5	Static-to-Deformed line appearance
6	Orbit trajectory time range

A5.4B.3 Card No. 3

1 10	
WINDOW	

(F10.0)

Window - The size of the shaft appearing on the plot.

A5.4B.4 Card No. 4

This card is used only if the third control parameter on card number 2 is non-zero.

1 10	
DISPSZ	

(F10.0)

DISPSZ - The size of the maximum orbit trajectory.

A5.4B.5 Card No. 5

This card is used only if the sixth control parameter on card number 2 is non-zero.

1 10	11 20	
PERMIN	PERMAX	

(2F10.0)

PERMIN - The minimum period at which the orbit trajectory is to be plotted.

PERMAX - The maximum period at which the orbit trajectory is to be plotted.

Note: TIMIN = PERIOD*PERMIN

TIMAX = PERIOD*PERMAX

A5.5 Program Listing, Post-Processors

ORIGINAL PAGE IS
OF POOR QUALITY

LIST UTILITY

```
//NASAPR      JOB  04180,NASA,CLASS=A
//   EXEC   FORT
C*************************************************************************
C*                                                                       *
C*     TITLE :  COMPUTER PLOTTING OF TRANSIENT AND STEADY-STATE          *
C*                 ROTOR VIBRATION TIME VARYING DEFLECTIONS              *
C*                                                                       *
C*                                                                       *
C*   PURPOSE :  THIS PROGRAM IS INTENDED TO SERVE AS THE FIRST OF TWO    *
C*                 PLOTTING POST-PROCESSOR PROGRAMS.  THE TASK OF THESE  *
C*                 PROGRAMS IS TO VISUAL DEPICT THE DEFORMATION PROCESS  *
C*                 OF A ROTATING SHAFT.  THIS POST-PROCESSOR PROGRAM     *
C*                 SERVES TO COMPUTE THE TIME VECTOR AT WHICH THE        *
C*                 DEFORMED SHAFT WILL BE PLOTTED AS WELL AS THE         *
C*                 GENERATION OF THE COEFFICIENTS FOR THE INTERPOLATORY  *
C*                 POLYNOMIAL THAT WILL BE USED TO DRAW THE SHAFT AT     *
C*                 EACH OF THE SELECTED TIME INTERVALS.                  *
C*                                                                       *
C*      NOTE :  FOR FURTHER DISCUSSION OF THIS PLOTTING PACKAGE,         *
C*                 PLEASE SEE THE REPORT RELATING TO THESE               *
C*                 POST-PROCESSOR PROGRAMS.                              *
C*                                                                       *
C*************************************************************************
      DIMENSION II(50), TIME(1000)
      COMMON/NUMBER/NBRPLT, NUMNP
      COMMON/PLTCNT/IPLOT , TIMINC, PLTTIM(1000)
      DATA IX   ,IY   ,IZ   /
     +     1HX  ,1HY  ,1HZ  /
C
C  INITIALIZE THE PLOT CONTROL COUNTER
C
      IPLOT=1
C
C  OBTAIN THE ADINA INFORMATION FROM THE DATA SET THAT WAS CREATED DURING
C     THE ADINA RUN.
C
      CALL RDADIN(PERIOD)
C
C  NOW READ THE PLOT TIME CONTROL DATA AND GENERATE THE PLOT TIME
C     VECTOR.  THIS VECTOR CONTAINS THE TIMES AT WHICH THE DEFORMED
C     SHAPE OF THE SHAFT IS TO BE VIEWED.
C
      READ(5,1)ANGINT,ANGFIN,ANGINC
    1 FORMAT(3F10.0)
      NBRPLT=0
      ANGLE=ANGINT
      IF(ANGINC .GT. 0)GO TO 3
      WRITE(6,2)
    2 FORMAT(' ****  ANGINC IS EITHER ZERO OR NEGATIVE.  *****'/
     +  '       SORRY THIS IS A FATAL ERROR.')
      STOP
    3 CONTINUE
      ANGLE=ANGLE+ANGINC
      NBRPLT=NBRPLT+1
      IF(ANGLE .LT. ANGFIN)GO TO 3
      IF(NBRPLT .LE. 1000)GO TO 5
      WRITE(6,4)
    4 FORMAT('     ****  NBRPLT EXCEEDS ITS MAXIMUM ALLOWABLE VALUE.  ***
```

LIST UTILITY

```
     +*'/.
     +.          '        SORRY FATAL ERROR.')
      STOP
    5 CONTINUE
      PLTTIM(1)=ANGINT*PERIOD
      TIMINC=ANGINC*PERIOD
      DO 6 I=2,NBRPLT
      PLTTIM(I)=PLTTIM(I-1)+TIMINC
    6 CONTINUE
C
C IDENTIFY, BY USER INPUT, THE SHAFT DIRECTION.
C
      READ(5,10)ID
   10 FORMAT(A1)
      ISHAFT=0
      IF(ID .EQ. IX)ISHAFT=1
      IF(ID .EQ. IY)ISHAFT=2
      IF(ID .EQ. IZ)ISHAFT=3
      IF(ISHAFT .NE. 0)GO TO 30
      WRITE(6,20)ID
   20 FORMAT('1 UNKNOWN ID INPUT, ID = ',A1)
      STOP 20
   30 CONTINUE
C
C SUBROUTINE TRAJ WILL SET-UP THE DATASETS FOR THE TRAJECTORY
C   PLOTTING.
C
      CALL TRAJ(II, AMAX, ISHAFT)
C
C WRITE OUT THE ADINA INFORMATION THAT WILL BE REQUIRED IN THE PLOTTING
C   ROUTINE.
C
      CALL WTADIN(II, ISHAFT, AMAX, PERIOD)
C
C DO THE FOLLOWING FOR EACH PLOTTING WINDOW.
C
      DO 50 I=1,NBRPLT
C
C READ THE ADINA DISPLACEMENTS FOR THIS TIME WINDOW.
C
      CALL RDDISP(ITIME, TIME(I))
C
C AT THIS POINT IF ITIME IS EQUAL ZERO, THEN WE HAVE NO DATA TO PLOT.
C
      IF(ITIME .EQ. 0)GO TO 49
C
C WRITE THE TIME OF THIS WINDOW TO THE STORAGE DATA SET FOR RETRIEVAL
C    DURING THE EXECUTION OF THE PLOTTING ROUTINE.
C
      WRITE(21,40)TIME(I)
      WRITE(6,40)TIME(I)
   40 FORMAT(2X,E13.6)
C
C COMPUTE AND OUTPUT THE COEFFICIENTS OF A FORWARD-DIFFERENCE NEWTON'S
C    INTERPOLATORY POLYNOMIAL FOR THIS TIME WINDOW.
C
      CALL COMDIF(ISHAFT)
   49 CONTINUE
```

LIST UTILITY

```
   50 CONTINUE
C
C HOPEFULLY, ALL DONE.
C
      STOP
      END
      SUBROUTINE COMDIF(ISHAFT)
C
C THIS SUBROUTINE COMPUTES AND OUTPUTS THE COEFFICIENTS OF A NEWTON'S
C   FORWARD-DIFFERENCE INTERPOLATORY POLYNOMIAL.  THESE COEFFICIENTS
C   ARE READ AND USED BY THE PLOTTING ROUTINE.  THERE IS A SET OF
C   COEFFICIENTS GENERATED FOR EACH TIME WINDOW TO BE PLOTTED.
C
      COMMON/DISPLA/DISP(3,50)
      COMMON/NODE  /LOC(3,50)
      COMMON/NUMBER/NBRPLT,NUMNP
      REAL          LOC
      DIMENSION     DIF(50,50)
      DO 70 IDIR=1,3
      IF(IDIR .EQ. ISHAFT)GO TO 60
      N=NUMNP-1
      DO 10 I=1,N
      DIF(1,I)=(DISP(IDIR,I+1)-DISP(IDIR,I))/(LOC(I+1,ISHAFT)
     *                                         - LOC(I,ISHAFT))
   10 CONTINUE
      DO 30 I=2,N
      K=NUMNP-I
      DO 20 L=1,K
      M=I+L
      DIF(I,L)=(DIF(I-1,L+1)-DIF(I-1,L))/(LOC(M,ISHAFT)
     *                                 - LOC(L,ISHAFT))
   20 CONTINUE
   30 CONTINUE
      WRITE(21,40)IDIR
      WRITE(6,40)IDIR
   40 FORMAT(2X,I3)
      WRITE(21,50)DISP(IDIR,1),(DIF(K,1),K=1,N)
      WRITE(6,50)DISP(IDIR,1),(DIF(K,1),K=1,N)
   50 FORMAT((5(2X,E13.6)/))
   60 CONTINUE
   70 CONTINUE
      RETURN
      END
      SUBROUTINE ROADIN(PERIOD)
      COMMON/NODE  /LOC(3,50)
      COMMON/NUMBER/NBRPLT,NUMNP
      REAL LOC
C
C READ THE NUMBER OF NODES
C
      READ(20,10)NUMNP
      WRITE(6,10)NUMNP
   10 FORMAT(2X,I10)
C
C READ THE X, Y, AND Z LOCATIONS OF THE NODES
C
      DO 30 I=1,NUMNP
      READ(20,20)(LOC(I,K),K=1,3)
```

LIST UTILITY

```
      WRITE(6,20)(LOC(I,K),K=1,3)
   20 FORMAT(3(2X,E15.8))
   30 CONTINUE
C
C READ THE PERIOD OF SHAFT REVOLUTION.
C
      READ(20,40)PERIOD
      WRITE(6,40)PERIOD
   40 FORMAT(2X,E15.7)
      RETURN
      END
      SUBROUTINE RDDISP(ITIME,TIME)
      COMMON/DISPLA/DISP(3,50)
      COMMON/NUMBER/NBRPLT,NUMNP
      COMMON/PLTCNT/IPLOT, TIMINC, PLTTIM(1000)
      DIMENSION II(50)
      ITIME=0
    4 CONTINUE
      READ(20,10,END=40)II(1),TIME,(DISP(K,1),K=1,3)
      DO 5 LL=IPLOT,NBRPLT
      IF(ABS(PLTTIM(LL)-TIME) .LE. TIMINC)GO TO 6
    5 CONTINUE
      DO 11 K=2,NUMNP
      READ(20,10)IDUM
   10 FORMAT(I10,4E13.6)
   11 CONTINUE
      GO TO 4
    6 CONTINUE
      ITIME=1
      IPLOT=IPLOT+1
      DO 20 I=2,NUMNP
      READ(20,10)II(I),TIME,(DISP(K,I),K=1,3)
   20 CONTINUE
      DO 30 I=1,NUMNP
      WRITE(6,10)II(I),TIME,(DISP(K,I),K=1,3)
   30 CONTINUE
      GO TO 60
   40 CONTINUE
      WRITE(6,50)
   50 FORMAT('     ******    END OF ADINA INFORMATION    *****     ')
      STOP
   60 CONTINUE
      RETURN
      END
      SUBROUTINE TRAJ(II, AMAX, ISHAFT)
      COMMON/NUMBER/NBRPLT,NUMNP
      DIMENSION II(50),TRAJD(50,2),TRAJA(50,3)
      REWIND 20
C
C READ UNWANTED DATA AT THE BEGINNING OF THE FILE.
C
      READ(20,1)IDUM
    1 FORMAT(2X,I10)
      DO 3 I=1,NUMNP
      READ(20,2)DUM
    2 FORMAT(2X,E15.8)
    3 CONTINUE
      READ(20,2)DUM
```

LIST UTILITY

```
      AMAX=0.0
    5 CONTINUE
      DO 20 I=1,NUMNP
      READ(20,10,END=40)II(I),TIME,(TRAJA(I,K),K=1,3)
   10 FORMAT(I10,4E13.6)
      IF(ISHAFT .NE. 1)GO TO 11
      TRAJD(I,1)=TRAJA(I,2)
      TRAJD(I,2)=TRAJA(I,3)
      GO TO 19
   11 CONTINUE
      IF(ISHAFT .NE. 2)GO TO 12
      TRAJD(I,1)=TRAJA(I,1)
      TRAJD(I,2)=TRAJA(I,3)
      GO TO 19
   12 CONTINUE
      TRAJD(I,1)=TRAJA(I,1)
      TRAJD(I,2)=TRAJA(I,2)
   19 CONTINUE
C
C COMPUTE THE MAGNITUDE OF THE TRAJECTOR PATH FOR THIS DISPLACEMENT
C     DATA.
C
      RANGE=SQRT(TRAJD(I,1)**2+TRAJD(I,2)**2)
C
C CHECK THIS MAGNITUDE AGAINST THE MAXIMUM AND IF THIS IS LARGER
C     SET THE MAXIMUM EQUAL TO THIS MAGNITUDE.
C
      IF(RANGE .GT. AMAX)AMAX=RANGE
   20 CONTINUE
      DO 30 I=1,NUMNP
      IOUT=10+I
      WRITE(IOUT,10)II(I),TIME,(TRAJD(I,K),K=1,2)
   30 CONTINUE
      GO TO 5
   40 CONTINUE
      REWIND 20
C
C READ UNWANTED DATA AT THE BEGINNING OF THE FILE.
C
      READ(20,1)IDUM
      DO 50 I=1,NUMNP
      READ(20,2)DUM
   50 CONTINUE
      READ(20,2)DUM
      RETURN
      END
      SUBROUTINE WTADIN(II, ISHAFT, AMAX, PERIOD)
      COMMON/NODE  /LOC(3,50)
      COMMON/NUMBER/NBRPLT,NUMNP
      DIMENSION II(50)
      REAL LOC
      WRITE(21,10)NUMNP,NBRPLT,ISHAFT,PERIOD
      WRITE(6,10)NUMNP,NBRPLT,ISHAFT,PERIOD
   10 FORMAT(3(2X,I10),E15.7)
      WRITE(21,11)(II(I),I=1,NUMNP)
   11 FORMAT((40(1X,I4)/))
      WRITE(21,15)AMAX
      WRITE(6,15)AMAX
```

```
LIST UTILITY
   15 FORMAT(2X,E15.8)
      WRITE(21,20)(LOC(I,ISHAFT),I=1,NUMNP)
      WRITE(6,20)(LOC(I,ISHAFT),I=1,NUMNP)
   20 FORMAT((5(2X,E13.6)/))
      RETURN
      END
```

```
C*********************************************************************
C*                                                                   *
C*                                                                   *
C*        TITLE: COMPUTER PLOTTING OF TRANSIENT AND STEADY-STATE     *
C*                  ROTOR VIBRATION TIME VARYING ANALYSIS            *
C*                                                                   *
C* DESCRIPTION: THIS PROGRAM IS THE SECOND OF TWO PROGRAMS THAT      *
C*                  ALLOW THE USER TO VIEW THE DEFORMATION OF A      *
C*                  ROTATING SHAFT.  THIS PROGRAM CANNOT BE          *
C*                  EXECUTED UNTIL THE FIRST ROUTINE HAS BEEN RUN    *
C*                  AND THE DATASETS THAT IT CREATES ARE ACCESSIBLE  *
C*                  TO THIS PROGRAM.  THIS PROGRAM PERFORMS THE      *
C*                  THE ACTUAL GRAPHICS ON THE CALCOMP PLOTTER.      *
C*                                                                   *
C*                                                                   *
C*********************************************************************
      COMMON/COEF   /A(2,50)  , LOC(50)
      COMMON/CNTRCL/ICNTRL(10)
      COMMON/DRAWX  /X(2,101)
      COMMON/NUM    /NUMNP    , NBRPLT
      COMMON/SCALE1/FACTRA    , FACTRB
      COMMON/TRIG   /COSINE   , SINE
      COMMON/TRJCNT/PERIOD    , TIMIN      , TIMAX
      DIMENSION ICIR(2)       , II(50)     , AXIAL(101)    ,XI(50)
      REAL LOC
      DATA PI/3.1415926536/,DISPSZ/1.5/,WINDOW/8.0/
      XMAX=0.0
      XMIN=0.0
C
C READ THE FOLLOWING INFORMATION FROM THE DATASET CREATED BY THE FIRST
C     POST-PROCESSOR:
C                  1)   NUMNP = NUMBER OF NODAL POINTS ON THE SHAFT.
C                  2) NBRPLT = NUMBER OF POSSIBLE PLOT TIMES.
C                  3) ISHAFT = ID OF THE SHAFT AXIAL COORDINATE.
C                  4) PERIOD = PERIOD OF SHAFT REVOLUTION.
C
      READ(21,10)NUMNP,NBRPLT,ISHAFT,PERIOD
   10 FORMAT(3(2X,I10),E15.7)
C
C READ THE ID'S OF THE SHAFT NODES.
C
      READ(21,20)(II(I),I=1,NUMNP)
   20 FORMAT((40(1X,I4)/))
      DO 30 I=1,NUMNP
      XI(I)=FLOAT(II(I))
   30 CONTINUE
C
C READ THE MAXIMUM-MAGNITUDE OF THE TRAJECTORY.
C
      READ(21,40)AMAX
   40 FORMAT(2X,E15.8)
C
```

LIST UTILITY

```
C READ THE AXIAL LOCATION OF THE NODES.
C
      READ(21,50)(LOC(I),I=1,NUMNP)
   50 FORMAT((5(2X,E13.6)/))
C
C SEARCH THE SHAFT FOR ITS MAXIMUN AND MINIMUM LOCATION.
C
      DO 60 I=1,NUMNP
      IF(LOC(I) .LT. XMIN)XMIN=LOC(I)
      IF(LOC(I) .GT. XMAX)XMAX=LOC(I)
   60 CONTINUE
C
C READ THE NUMBER OF SECTIONS INTO WHICH THE SHAFT IS TO BE DIVIDED.
C
      READ(5,70)NSECT,THETA
   70 FORMAT(I5,F10.0)
C
C READ THE VALUES OF THE CONTROL VECTOR.  THE VALUES READ WILL CONTROL
C      THE FOLLOWING ITEMS:
C        ICNTRL
C       COMPONENT        CONTROLS
C           1            THE NUMBER OF DEFORMED SHAFTS THAT ARE PLOTTED IN
C                            EACH WINDOW.
C           2            THE USEAGE OF THE DEFAULT PLOTTING WINDOW SIZE.  IF
C                            THE VALUE READ IS ZERO, THEN THE DEFAULT SIZE
C                            8 INCHES IS USED.  IF THE VALUE READ IS ONE,
C                            THEN THIS WINDOW SIZE MUST BE INPUT.
C           3            THE USEAGE OF THE DEFAULT AMPLIFICATION FACTOR.
C                            IF THE VALUE READ IS ZERO, THEN THE AMPLI -
C                            FICATION FACTOR USED WILL PRODUCE A MAXIMUM
C                            TRAJECTORY ORBIT OF 1.5 INCHES.  IF THE VALUE
C                            READ IS ONE, THEN THE SIZE OF THE MAXIMUM
C           4            THE APPEARENCE OF THE NODE NAMES ON THE PLOT.
C                            IF THE VALUE READ IS ZERO, THEN THE NAMES
C                            APPEAR.  IF THE VALUE READ IS ONE THEN THE
C                            DO NOT APPEAR.
C           5            THE APPEARENCE OF THE LINE CONNECTING THE STATIC
C                            POSITIONS OF THE NODES TO THEIR DEFORMED
C                            LOCATIONS.  IF THE VALUE READ IS ZERO, THEN THIS
C                            LINE APPEARS.  IF THE VALUE READ IS ONE, THEN
C                            THIS LINE DOES NOT APPEAR.
C           6            THE TIME RANGE OF THE TRAJECTORY ORBIT THAT IS
C                            OUTPUT IN EACH PLOTTING WINDOW.  IF THE VALUE
C                            READ IS ZERO, THEN THE ENTIRE ORBIT IS DISPLAYED
C                            IN EACH PLOTTING WINDOW.  IF THE VALUE READ IS
C                            ONE, THEN ONLY THE RANGE SPECIFIED BY THE USER
C                            ISPLOTTED.
C
      READ(5,80)(ICNTRL(I),I=1,10)
   80 FORMAT(I5,9I1)
      WRITE(6,83)(ICNTRL(I),I=1,10)
   83 FORMAT(' ICNTRL = ',10(I3,2X)////)
C
C DEFALT WILL USE THE VALUES OF THE CONTROL VECTOR TO DETERMINE IF ANY
C     ADDITIONAL INPUTS MUST BE READ AND IT WILL THEN READ THESE INPUTS.
C
      CALL DEFALT(DISPSZ, WINDOW)
C
```

```
LIST UTILITY
C COMPUTE THE INCREMENTAL AXIAL DISTANCE.
C
      DDIST=(XMAX-XMIN)/FLOAT(NSECT)
C
C CONVERT THETA TO RADIANS FROM DEGREES.
C
      THETA=THETA*PI/180.
C
C TO AVOID MULTIPLE CALLS TO THE TRIG FUNCTIONS, COMPUTE THE SINE
C     AND COSINE OF THETA.
C
      COSINE=COS(THETA)
      SINE=SIN(THETA)
C
C FIND THE MAXIMUM X1 RANGE.
C
      XRNGE=ABS(XMAX*COSINE-XMIN*COSINE)
C
C FIND THE MAXIMUM X2 RANGE.
C
      YRNGE=ABS(XMAX*SINE-XMIN*SINE)
C
C DETERMINE THE MAXIMUM RANGE OF THE PROBLEM.
C
      IF(XRNGE .GE. YRNGE)RANGE=XRNGE
      IF(YRNGE .GT. XRNGE)RANGE=YRNGE
C
C WITH THE MAXIMUM RANGE AND THE SIZE OF THE VIEWING WINDOW, COMPUTE
C     THE WINDOW SCALING FACTOR.
C
      FACTRB=WINDOW/RANGE
C
C WITH THE MAXIMUM MAGNITUDE OF THE TRAJECTORY, COMPUTE THE DISPLACEMENT
C      SCALING FACTOR.
C
      FACTRA=DISPSZ/AMAX
C
C INITIALIZE THE PLOTTER.
C
      CALL PLOTS
C
C MOVE AND CENTER THE ORIGIN OF THE PLOTTER
C
      CENTER=0.5*(11.0-WINDOW)
      CALL PLOT(0.0, CENTER, -3)
C
C RESET THE NUMBER OF SECTIONS.
C
      NSECT=NSECT+1
      JJ=ICNTRL(1)
      IF(JJ .EQ. 0)JJ=1
C
C DO THE FOLLOWING FOR ALL THE POSSIBLE PLOT TIMES.
C
      DO 130 LL=1,NBRPLT
      DO 120 KK=1,JJ
C
C READ THE PLOT TIME.
```

ORIGINAL PAGE IS
OF POOR QUALITY

LIST UTILITY

```
C
      READ(21,50,END=140)TIME
C
C DO THE FOLLOWING FOR THE DEFLECTION DIRECTIONS.
C
      DO 90 I=1,2
C
C READ THE DIRECTION ID.
C
      READ(21,85)IDIR(I)
   85 FORMAT(2X,I3)
C
C READ THE FORWARD NEWTON INTERPOLATORY POLYNOMIAL COEFFICIENTS.
C
      READ(21,50)(A(I,K),K=1,NUMNP)
   90 CONTINUE
C
C NOW FOR EACH LOCATION ALONG THE SHAFT.
C
      DO 100 I=1,NSECT
C
C DETERMINE THE AXIAL LOCATION OF THIS POINT.
C
      DIST=XMIN+(DDIST*(I-1))
C
C CALL DISP TO EVALUATE THE FORWARD NEWTON INTERPOLATORY POLYNOMIAL AT
C    THIS AXIAL LOCATION.
C
      CALL DISP(DIST, X1, X2)
C
C STORE THESE COORDINATES IN THE COORDINATE MATRIX.
C
      AXIAL(I)=DIST
      X(1,I)=X1
      X(2,I)=X2
  100 CONTINUE
      DO 110 I=1,NSECT
C
C COMPUTE THE SCALED DISPLACEMENTS.
C
      X(1,I)=X(1,I)*FACTRA
      X(2,I)=X(2,I)*FACTRA
C
C COMPUTE THE SCALED LOCATIONS.
C
      AXIAL(I)=AXIAL(I)*FACTRB
C
C NOW PLACE THE SCALED DISPLACEMENTS INTO THE PROPER GLOBAL
C    VIEWING PERSPECTIVE.
C
      CALL COORD(AXIAL(I), X(1,I), X(2,I))
  110 CONTINUE
C
C DRAW THE UNDEFORMED, STATIC SHAFT.
C
      CALL STATIC
C
C CALL DRAW TO PLOT OUT THE DEFLECTION SHAPE OF THE SHAFT AT THIS TIME.
```

LIST UTILITY

```
C
      CALL DRAW(NSECT)
C
C CONNECT THE UNDEFORMED NODAL LOCATIONS WITH THEIR DEFORMED LOCATIONS.
C
      IF(ICNTRL(5) .EQ. 0)CALL LINE
C
C OUTPUT TO THE PRINTER THE PLOT TIME JUST PLOTTED.
C
      WRITE(6,115)TIME
  115 FORMAT('    PLOT DRAWN FOR TIME = ',E15.8)
C
C IF MULTIPLE SHAFTS ARE TO APPEAR ON THE SAME PICTURE, THEN THERE IS
C    NO NEED TO REDRAW THE REMAINING COMPONENTS EACH TIME.
C
      IF(KK .NE. 1)GO TO 120
C
C PLOT OUT THE SHAFT NODAL TRAJECTORIES.
C
      CALL TRJPLT(II)
C
C WRITE OUT THE NAMES OF THE NODES.
C
      IF(ICNTRL(4) .EQ. 0)CALL NAME(CENTER, XI)
C
C PRINT OUT THE TIME ASSOCIATED WITH THIS PLOT.
C
      CALL SYMBOL(3.0, 1.0, 0.25, 7HTIME =  , 0.0, 7)
      CALL NUMBER(999., 999., 0.25, TIME, 0.0, 4)
  120 CONTINUE
C
C MOVE THE PLOT ORIGIN FOR THE NEXT WINDOW IN TIME.
C
      CALL PLOT(17.0, 0.0, -3)
  130 CONTINUE
  140 CONTINUE
C
C HOPEFULLY, ALL DONE.
C
      CALL PLOT(20.0, 0.0, 999)
      STOP
      END
      SUBROUTINE COORD(DIST, X1, X2)
      COMMON/TRIG  /COSINE, SINE
      X1=X1 + DIST*COSINE
      X2=X2 + DIST*SINE
      RETURN
      END
      SUBROUTINE DEFALT(DISPSZ, WINDOW)
      COMMON/CNTRCL/ICNTRL(10)
      COMMON/TRJCNT/PERIOD, TIMIN  , TIMAX
      IF(ICNTRL(2) .EQ. 0)GO TO 20
      READ(5,10)WINDOW
   10 FORMAT(F10.0)
   20 CONTINUE
      IF(ICNTRL(3) .EQ. 0)GO TO 30
      READ(5,10)DISPSZ
   30 CONTINUE
```

LIST UTILITY

```
      IF(ICNTRL(6) .EQ. 0)GO TO 50
      READ(5,40)PERMIN,PERMAX
   40 FORMAT(2F10.0)
      TIMIN=PERMIN*PERIOD
      TIMAX=PERMAX*PERIOD
      GO TO 60
   50 CONTINUE
      TIMIN=0.0
      TIMAX=1.E+10
   60 CONTINUE
      WRITE(6,70)TIMIN,TIMAX
   70 FORMAT('      TIMIN = ',E15.7/
     +       '      TIMAX = ',E15.7)
      RETURN
      END
      SUBROUTINE DISP(DIST, X1, X2)
      COMMON/COEF   /A(2,50)  , LOC(50)
      COMMON/NUM    /NUMNP    , NERPLT
      DIMENSION A1(2,50)
      REAL LOC
C
C THIS SUBROUTINE WILL EVALUATE THE NEWTON FORWARD INTERPOLATORY
C    POLYNOMIAL THAT WILL RESULT IN THE APPROXIMATION OF THE SHAFT
C    DISPLACEMENT AT THE DISTANCE DIST.
C THE DISPLACEMENT IS DEFINED BY DETERMINING THE TWO COMPONENTS OF THE
C    SHAFT DISPLACEMENT.
C
      N=NUMNP-1
      DO 20 I=1,2
      DO 10 K=1,NUMNP
      A1(I,K)=A(I,K)
   10 CONTINUE
   20 CONTINUE
      DO 40 I=1,2
      DO 30 K=1,N
      L=NUMNP-K
      A1(I,L)=A1(I,L)+(DIST-LOC(L))*A1(I,L+1)
   30 CONTINUE
   40 CONTINUE
      X1=A1(1,1)
      X2=A1(2,1)
      RETURN
      END
      SUBROUTINE DRAW(NSECT)
      COMMON/COEF   /A(2,50)  , LOC(50)
      COMMON/DRAWX  /X(2,101)
      COMMON/NUM    /NUMNP    , NERPLT
      COMMON/SCALE1/FACTRA    , FACTRB
      REAL          LOC
      DO 15 J=1,2
      X1=X(1,1)
      X2=X(2,1)-(J-1)*0.015
      CALL PLOT(X1, X2, 3)
      DO 10 I=2,NSECT
      X1=X(1,I)
      X2=X(2,I)-(J-1)*0.015
      CALL PLOT(X1, X2, 2)
   10 CONTINUE
```

ORIGINAL PAGE IS
OF POOR QUALITY

LIST UTILITY

```
   15 CONTINUE
      DO 20 I=1,NUMNP
      AXIAL=LOC(I)
      CALL DISP(AXIAL, X1, X2)
      X1=X1*FACTRA
      X2=X2*FACTRA
      AXIAL=AXIAL*FACTRB
      CALL COORD(AXIAL, X1, X2)
      CALL SYMBOL(X1, X2, 0.1, 11, 0.0, -1)
      CALL SYMBOL(X1, X2, 0.1, 0, 0.0, -1)
   20 CONTINUE
      RETURN
      END
      SUBROUTINE LINE
      COMMON/COEF   /A(2,50), LOC(50)
      COMMON/NUM    /NUMNP  , NBRPLT
      COMMON/SCALE1/FACTRA  , FACTRB
      COMMON/TRIG   /COSINE , SINE
      REAL LOC
      DO 10 I=1,NUMNP
      X1=LOC(I)*COSINE*FACTRB
      X2=LOC(I)*SINE*FACTRB
      CALL PLOT(X1, X2, 3)
      CALL DISP(LOC(I), X1, X2)
      X1=X1*FACTRA
      X2=X2*FACTRA
      POS=LOC(I)*FACTRB
      CALL COORD(POS, X1, X2)
      CALL PLOT(X1, X2, 2)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE NAME(CENTER, XI)
      COMMON/COEF   /A(2,50), LOC(50)
      COMMON/NUM    /NUMNP  , NBRPLT
      COMMON/SCALE1/FACTRA  , FACTRB
      COMMON/TRIG   /COSINE , SINE
      REAL LOC
      DIMENSION XI(50)
      DO 10 I=1,NUMNP
      X1=LOC(I)*FACTRB*COSINE
      X2=LOC(I)*FACTRB*SINE
      CALL PLOT(X1, X2, 3)
      X1=X1+ 0.6*CENTER
      X2=X2-0.8*CENTER
      CALL PLOT(X1, X2, 2)
      X=XI(I)
      CALL NUMBER(X1, X2, 0.20, X, 0.0, -1)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE STATIC
      COMMON/COEF   /A(2,50), LOC(50)
      COMMON/NUM    /NUMNP  , NBRPLT
      COMMON/SCALE1/FACTRA  , FACTRB
      COMMON/TRIG   /COSINE , SINE
      REAL LOC
      RANGE=LOC(NUMNP)-LOC(1)
```

LIST UTILITY

```
      DDIST=RANGE/42.0
      DIST=LOC(1)
      DO 10 I=1,42,2
      X1=(DIST+(I-1)*DDIST)*COSINE*FACTRB
      X2=(DIST+(I-1)*DDIST)*SINE*FACTRB
      CALL PLOT(X1, X2, 3)
      X1=(DIST+I*DDIST)*COSINE*FACTRB
      X2=(DIST+I*DDIST)*SINE*FACTRB
      CALL PLOT(X1, X2, 2)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE TRJPLT
      COMMON/COEF   /A(2,50) , LOC(50)
      COMMON/NUM    /NUMNP    , NUMPLT
      COMMON/SCALE1/FACTRA , FACTRB
      COMMON/TRIG   /COSINE   , SINE
      COMMON/TRJCNT/PERIOD    , TIMIN    ,TIMAX
      REAL LOC
      HGT=0.05
      HGT2=0.100
      ISYM=50
      DO 30 I=1,NUMNP
      ISYM1=1
      IN=10+I
    1 CONTINUE
      READ(IN,5)II,TIME,TRAJ1,TRAJ2
    5 FORMAT(I10,3E13.6)
      IF(TIME .LT. TIMIN)GO TO 1
      X1=TRAJ1*FACTRA
      X2=TRAJ2*FACTRA
      POS=LOC(I)*FACTRB
      CALL COORD(POS, X1, X2)
      IF(X1 .GT. 1000. .OR. X2 .GT. 1000.)STOP 9900
      CALL PLOT(X1, X2, 3)
      CALL SYMBOL(X1, X2, HGT2, 1, 0.0, -1)
   10 CONTINUE
      READ(IN,5,END=20)II,TIME,TRAJ1,TRAJ2
      IF(TIME .GT. TIMAX)GO TO 20
      ISYM1=ISYM1+1
      X1=TRAJ1*FACTRA
      X2=TRAJ2*FACTRA
      POS=LOC(I)*FACTRB
      CALL COORD(POS, X1, X2)
      IF(X1 .GT. 1000. .OR. X2 .GT. 1000.)STOP 9900
      CALL PLOT(X1, X2, 2)
      IF(ISYM1 .NE. ISYM)GO TO 10
      ISYM1=1
      CALL SYMBOL(X1, X2, HGT, 1, 0.0, -1)
      GO TO 10
   20 CONTINUE
      REWIND IN
   30 CONTINUE
      RETURN
      END
```

750 RECORDS PRINTED. END OF LIST UTILITY

ORIGINAL PAGE IS
OF POOR QUALITY

A5.6 Sample Run of the 3-D Plotting Post-Processors

A5.6A ADINA Generated Data

The necessary data from the modified version of ADINA was obtained through Fortran logical unit 20. The JCL used to allocate this unit was

```
//FT20F001 DD DSN=USER.MULTI1,DISP=(NEW,CATLG),
//             SPACE=(TRK,(150,20),RLSE),
//             UNIT=SYSDA,VOL=SER=ACAD01,
//             DCB=(RECFM=FB,BLKSIZE=12960,LRECL=80)
```

A5.6B Post-Processor / JCL Requirements

The JCL required by the first post-processor program for this sample problem that contained three nodes along the shaft was:

```
//  EXEC  GOFORT
//FT11F001 DD DSN=USER.EFHPLT1 DISP=(NEW,CATLG),UNIT=SYSDA,
//              VOL=SER=ACAD01,SPACE=(TRK,(50,5),RLSE),
//              DCB=(RECFM=FB,BLKSIZE=6160,LRECL=80)
//FT12F001 DD DSN=USER.EFHPLT2,DISP=(NEW,CATLG),UNIT=SYSDA,
//              VOL=SER=ACAD01,SPACE=(TRK,(50,5),RLSE),
//              DCB=(RECFM=FB,BLKSIZE=6160,LRECL=80)
//FT13F001 DD DSN=USER,EFHPLT3,DISP=(NEW,CATLG),UNIT=SYSDA,
//              VOL=SER+ACAD01,SPACE=(TRK,(50,5),RLSE),
//              DCB=(RECFM=FB,BLKSIZE=6160,LRECL=80)
//FT20F001 DD DSN=USER,MULTI1,DISP=SHR
//FT21F001 DD DSN=USER,EFHPLOT,DISP=(NEW,CATLG),UNIT=SYSDA,
//              VOL=SER=ACAD01,SPACE=(TRK,(50,5),RLSE),
//              DCB=(RECFM=FB,BLKSIZE=6160,LRECL=80)
```

A5.6C Post-Processor / Input

The user inputs to the first post-processor program for this sample run was:

```
0.0     90.0     0.3
X
```

A5.6D Post-Processor 2, JCL Requirements

The JCL required by the second post-processor program for this sample was:

```
//  EXEC  GOFORT,GOSIZE=512K,PLOT=
//FT11F001 DD DSN=USER.EFHPLT1,DISP=SHR
//FT12F001 DD DSN=USER.EFHPLT2,DISP=SHR
//FT13F001 DD DSN=USER.EFHPLT3,DISP=SHR
//FT21F001 DD DSN=USER.EFHPLOT,DISP=SHR
```

A5.6E Post-Processor 2 Input

The user inputs for the second post-processor program for this sample run was:

```
 20    35
3
```

A5.6F Post-Processor 2 Output

Attached are some selected plots that were obtained during this sample run.

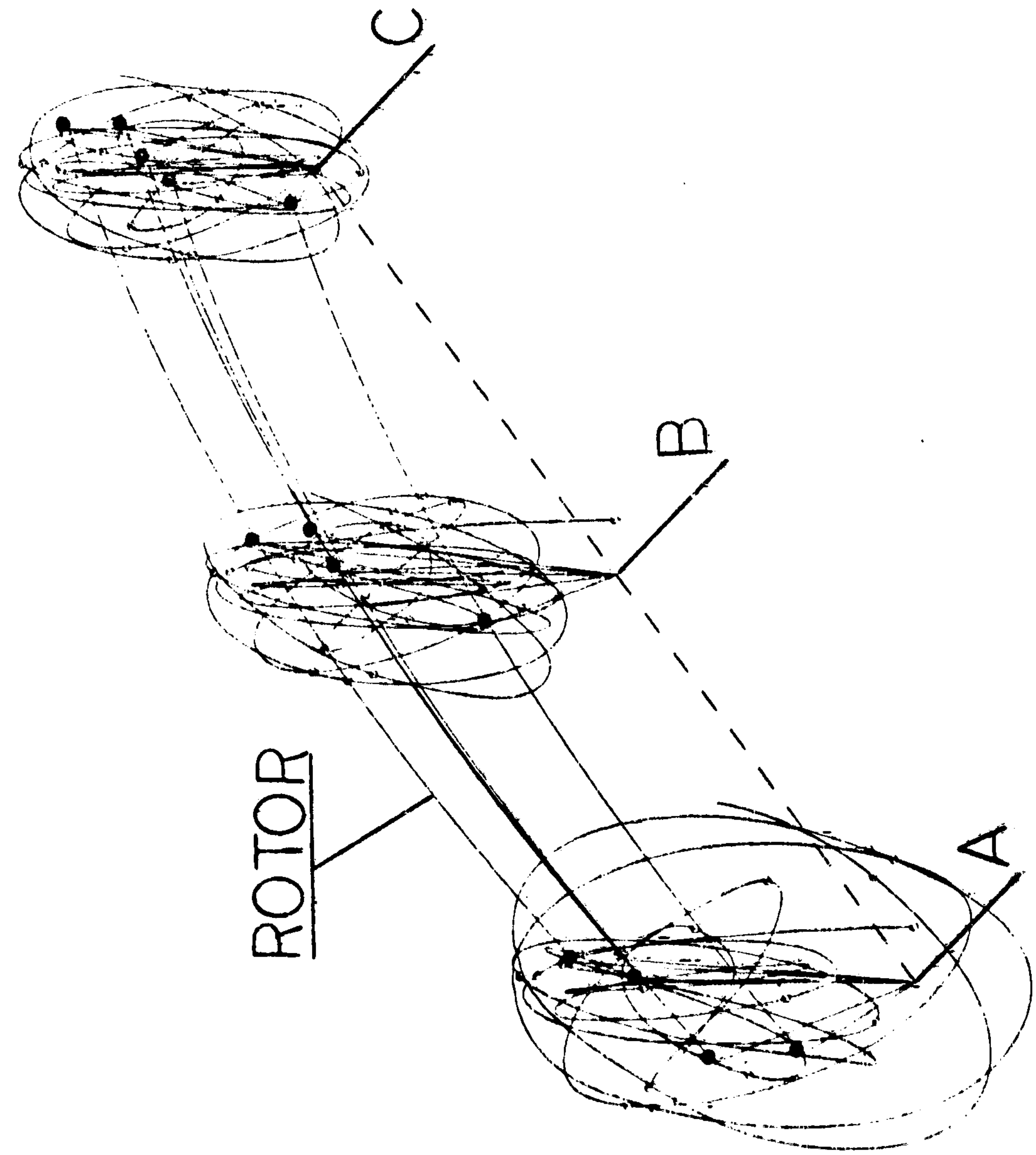


FIG.A5-5 ROTOR TRAJECTORY

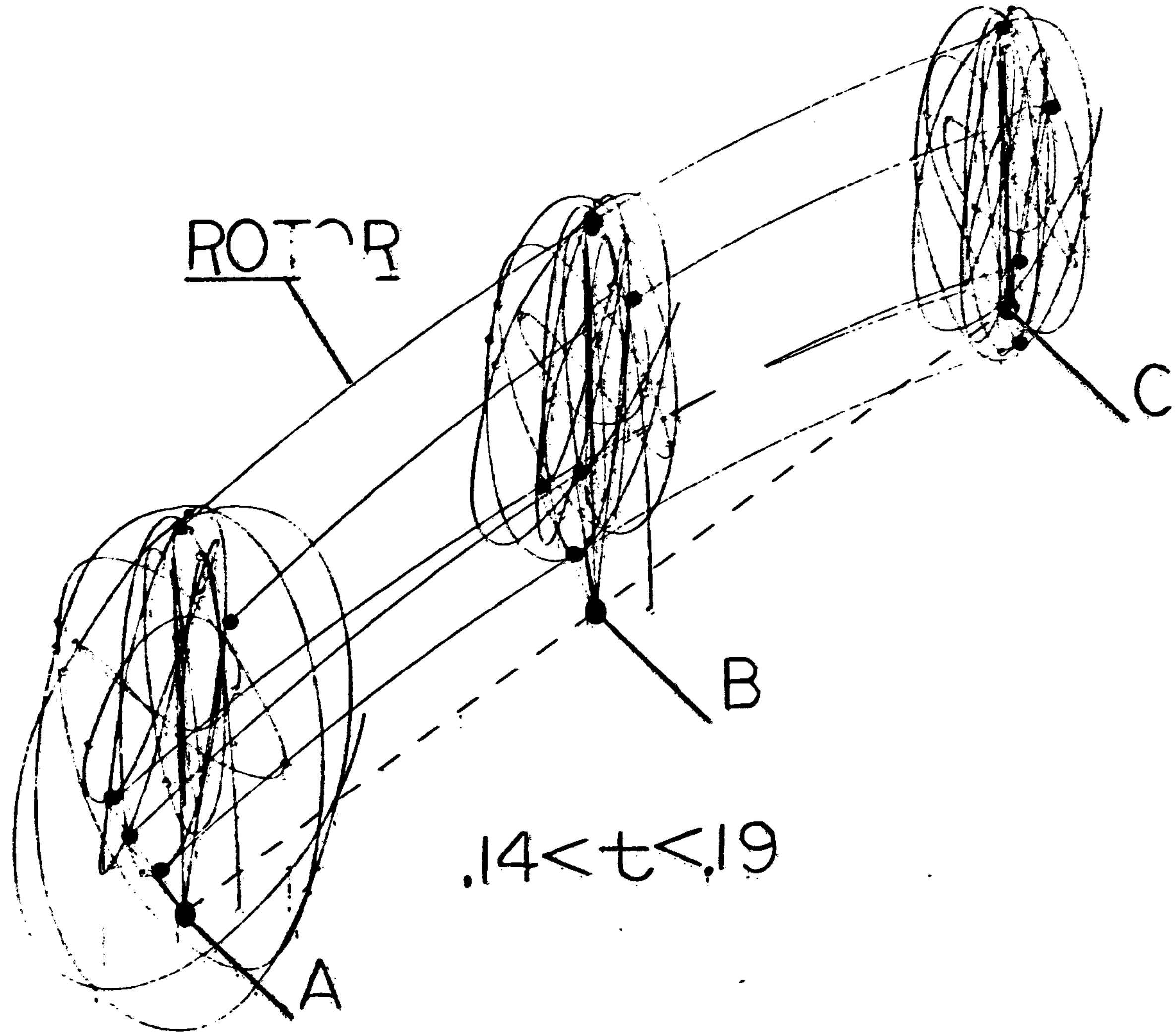


FIG A5-4 ROTOR TRAJECTORY

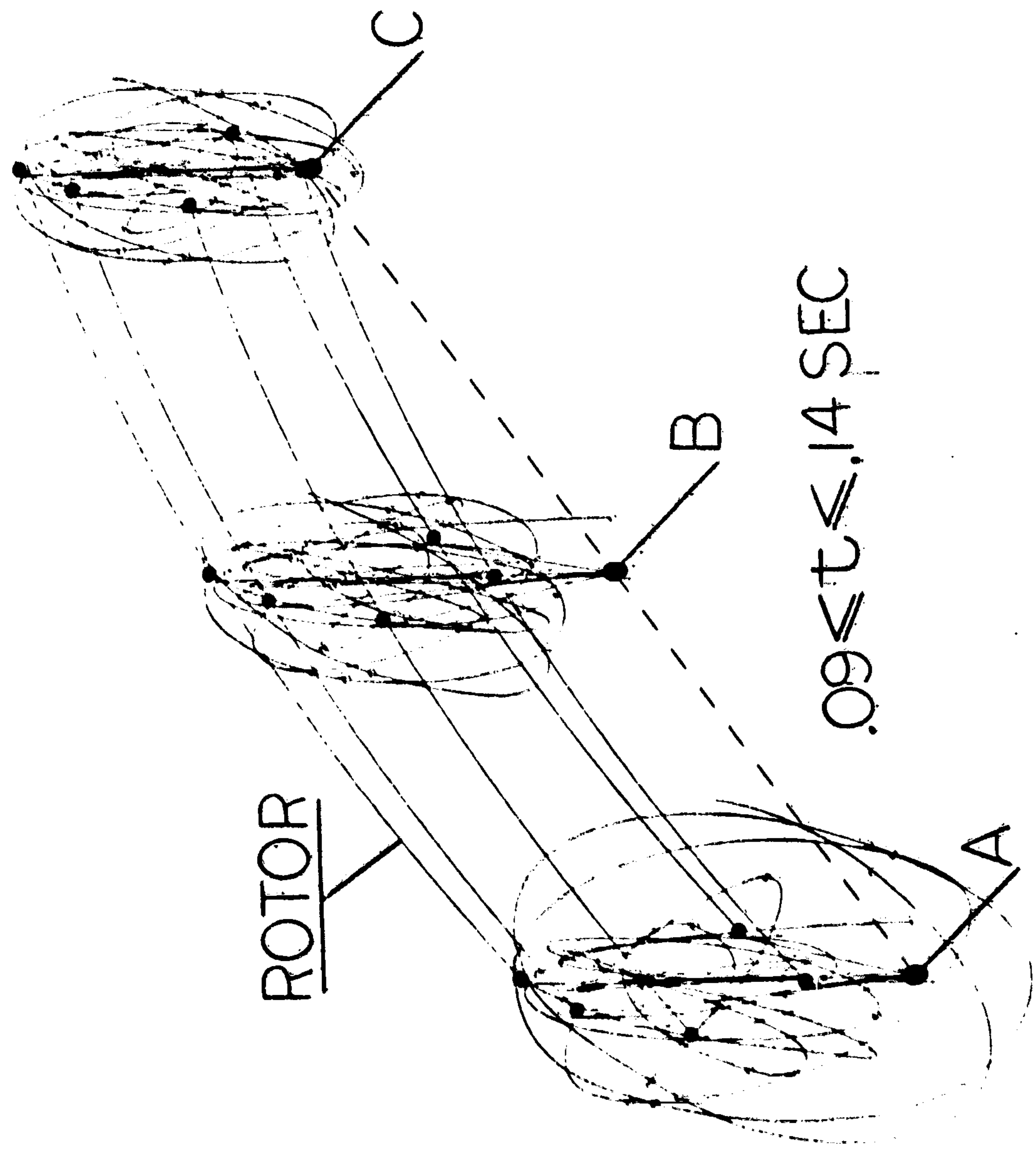


FIG.A5-3 ROTOR TRAJECTORY

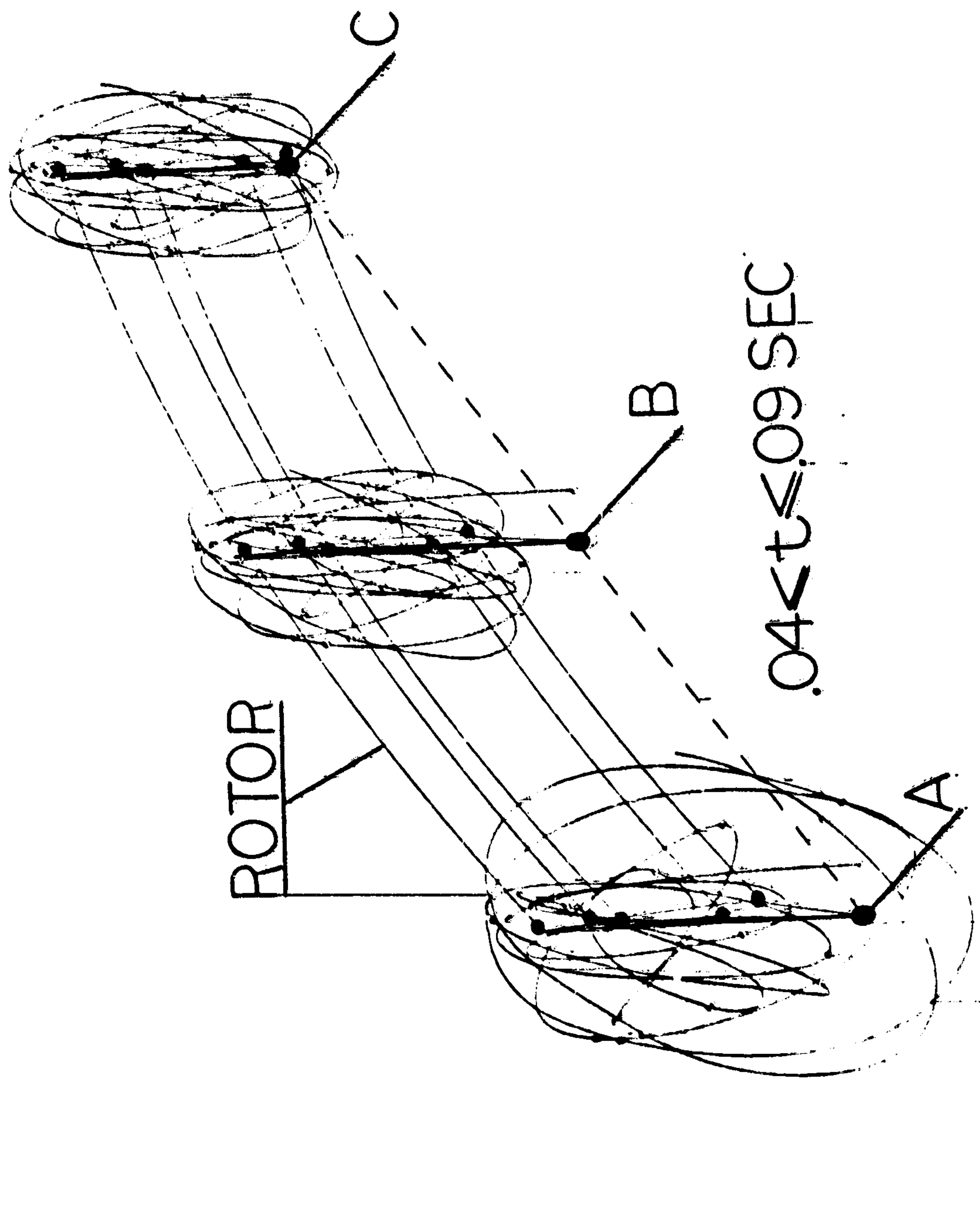


FIG A5-2 ROTOR TRAJECTORY

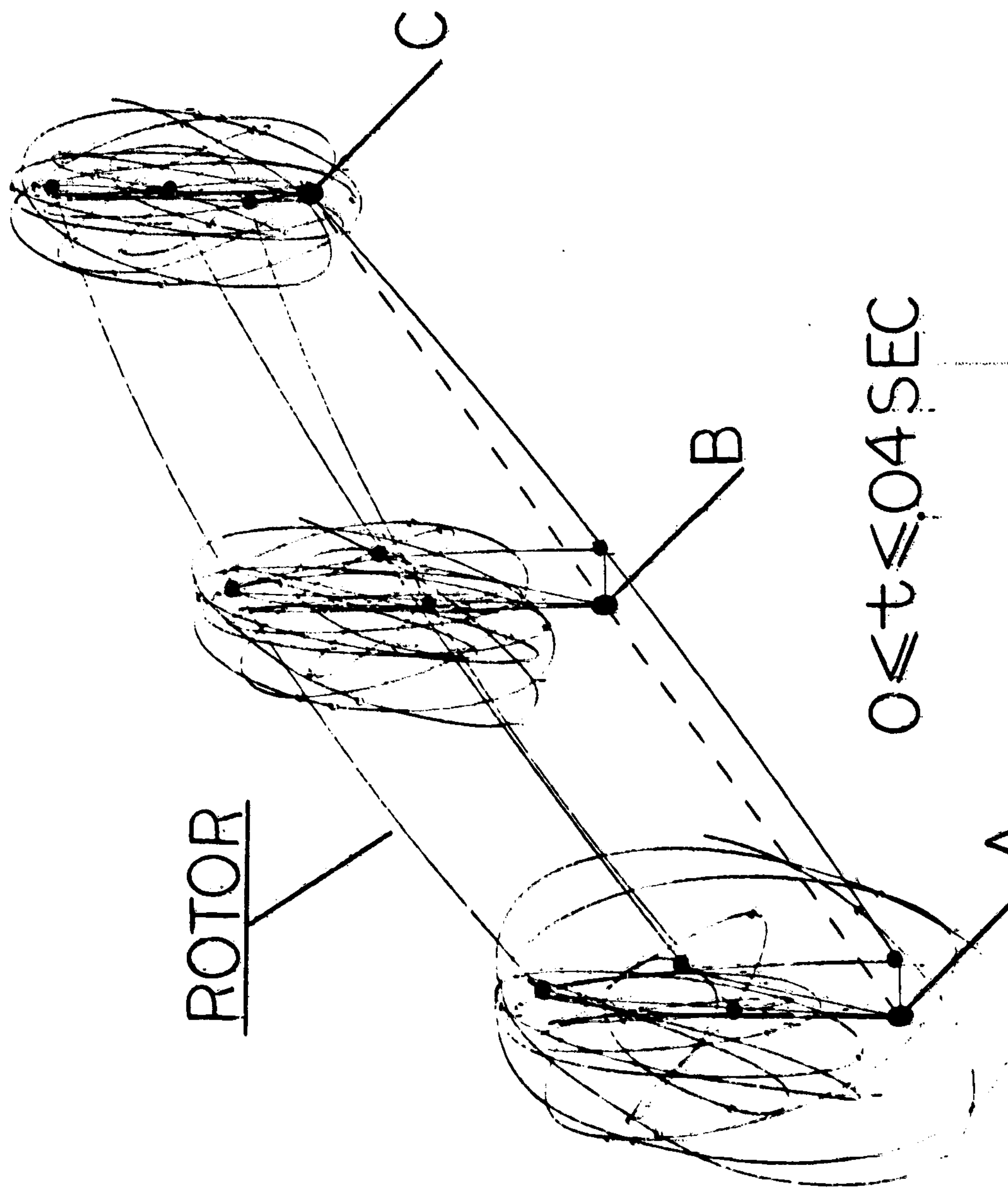


FIG.A5-1 ROTOR TRAJECTORY

APPENDIX 6: NOMENCLATURE

a_j - Cartesian Lagrangian Coordinates

[B*] - Matrix Coefficients
[B]

$[C_b]$ - Tangential bearing damping matrix

$\underset{\sim}{F}_e$ - Elemental nodal force matrix

$\underset{\sim}{F}_b$ -- Bearing nodal load

fe - Element body force

[K] - Global stiffness

$[K_e]$ - Element stiffness

$[K_{ep}]$ - Elastic-plastic stiffness

$[K_l]$ - Linearized stiffness

$[K_d]$ - Dynamic stiffness

L_{ij} - Lagrangian strain tensor
$\underset{\sim}{L}$

$[M_e]$ - Element mass matrix

[M] - Global mass matrix

[N] - Shape function

P_i, P_o - Inlet-outlet pressures of Damper Bearing

R - Region occuppied by system

S_{ij} - 2nd Piola-Kirchhoff Stress tensor
$\underset{\sim}{S}$

t - Time

U_i - Displacements
$\underset{\sim}{U}$

$\underset{\sim}{Y}$ - Nodal displacements

v - Volume

$\Delta\underset{\sim}{Y}$ - Incremental nodal displacement

Δt - Time increment

$\delta()$ - Variational operator

ρ_0 - Density

( ) - Column Matrix

[ ] - Matrix

$(\ )^T$ - Transpose

$(\dot{})$, $( )_{,t}$ - Time differentiation

$(\ )^T$ - Matrix Transpose

$\|\ \|$ - Euclidian Norm

APPENDIX 7: EQUATION DEFINITION

$$[B] = [[B_i], [B_j], \ldots]$$

$$[B_\ell] = \begin{bmatrix} N_{\ell,} & 0 & 0 \\ 0 & N_{\ell,2} & 0 \\ 0 & 0 & N_{\ell,2} \\ N_{\ell,2} & N_{\ell,1} & 0 \\ 0 & N_{\ell,3} & N_{\ell,2} \\ N_{\ell,3} & 0 & N_{\ell,1} \end{bmatrix} ;\ell = i, j, \ldots$$

$$[G] = [[G_i], [G_j], \ldots]$$

$$[G_\ell] = \begin{bmatrix} N_{\ell,1} & 0 & 0 \\ N_{\ell,2} & 0 & 0 \\ N_{\ell,3} & 0 & 0 \\ 0 & N_{\ell,1} & 0 \\ 0 & N_{\ell,2} & 0 \\ 0 & N_{\ell,3} & 0 \\ 0 & 0 & N_{\ell,1} \\ 0 & 0 & N_{\ell,2} \\ 0 & 0 & N_{\ell,3} \end{bmatrix} ;\ell = i, j, \ldots$$